Automating Measurements

Agilent Technologies 8712ET/ES and 8714ET/ES RF Network Analyzers

User's Guide Supplement

Agilent Technologies

Part No. 08714-90014

Printed in USA

Print Date: June 2000

Supersedes: October 1999

© Copyright 1998-2000 Agilent Technologies, Inc.

Notice

The information contained in this document is subject to change without notice. Agilent Technologies makes no warranty of any kind with regard to this material, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Key Conventions

This manual uses the following conventions:

(FRONT PANEL KEY): This represents a key physically located on the analyzer (a "hardkey").

Softkey : This indicates a "softkey"—a key whose label is determined by the instrument's firmware, and is displayed on the right side of the instrument's screen next to the eight unlabeled keys.

Acknowledgments

Excel™ is a product of Microsoft Corp.

Lotus® 1-2-3®, and Lotus WordPro® are U.S. registered trademarks of Lotus Development Corporation.

Microsoft is a U.S. registered trademark of Microsoft Corporation.

QuickBasic™ is a product of Microsoft Corp.

Portions of the software include source code from the Info–ZIP group. This code is freely available on the Internet by anonymous ftp asftp.uu.net:/pub/archiving/zip/unzip51/.tar.Z, and from CompuServe asunz51.zip in the IBMPRO forum, library 10, (data compression).

Agilent Technologies Sales & Service Offices

UNITED STATES

Instrument Support Center
Agilent Technologies, Inc.
(800) 403-0801

EUROPEAN FIELD OPERATIONS

Headquarters
Agilent Technologies S.A.
150, Route du Nant-d'Avril
1217 Meyrin 2/ Geneva
Switzerland
(41 22) 780.8111

France
Agilent Technologies France
1 Avenue Du Canada
Zone D'Activite De Courtaboeuf
F-91947 Les Ulis Cedex France
(33 1) 69 82 60 60

Germany
Agilent Technologies GmbH
Agilent Technologies Strasse
61352 Bad Homburg v.d.H
Germany
(49 6172) 16-0

Great Britain
Agilent Technologies Ltd.
Eskdale Road, Winnersh Triangle
Wokingham, Berkshire RG41 5DZ
England
(44 118) 9696622

INTERCON FIELD OPERATIONS

Headquarters
Agilent Technologies
3495 Deer Creek Rd.
Palo Alto, CA 94304-1316
USA
(650) 857-5027

Australia
Agilent Technologies Australia Ltd.
31-41 Joseph Street
Blackburn, Victoria 3130
(61 3) 895-2895

Canada
Agilent Technologies (Canada) Ltd.
17500 South Service Road
Trans-Canada Highway
Kirkland, Quebec H9J 2X8
Canada
(514) 697-4232

Japan
Agilent Technologies Japan, Ltd.
Measurement Assistance Center
9-1, Takakura-Cho, Hachioji-Shi
Tokyo 192-8510, Japan
TEL (81) -426-56-7832
FAX (81) -426-56-7840

Singapore
Agilent Technologies Singapore (Pte.) Ltd.
150 Beach Road
#29-00 Gateway West
Singapore 0718
(65) 291-9088

Taiwan
Agilent Technologies Taiwan
8th Floor, H-P Building
337 Fu Hsing North Road
Taipei, Taiwan
(886 2) 712-0404

China
China Agilent Technologies Co.
38 Bei San Huan X1 Road
Shuang Yu Shu
Hai Dian District
Beijing, China
(86 1) 256-6888

Agilent 8712ET/ES and 8714ET/ES Network Analyzer Documentation Map

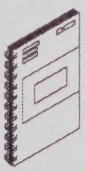
The **CDROM** provides the contents of all of the documents listed below.

The **User's Guide** shows how to make measurements, explains commonly-used features, and tells you how to get the most performance from the analyzer.

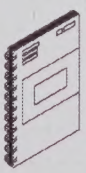
The **LAN Interface User's Guide Supplement** shows how to use a local area network (LAN) for programming and remote operation of the analyzer.

The **Automating Measurements User's Guide Supplement** provides information on how to configure and control test systems for automation of test processes.

The **Programmer's Guide** provides programming information including GPIB and SCPI command references, as well as short programming examples.



The **Example Programs Guide** provides a tutorial introduction using BASIC programming examples to demonstrate the remote operation of the analyzer.



1 Introduction

Automating Measurements

An automated measurement system is one where a computer performs some of the tasks that you would normally have to do manually.

The information in this book will help you learn how to automate your measurement system. Several features of the instrument that are useful for automation are explained.

NOTE

The information in this book assumes that you are familiar with the information provided in the *Programmer's Guide*, the *HP Instrument BASIC User's Handbook*, and the *HP Instrument BASIC User's Handbook Supplement* that were supplied with your analyzer.

This book explains how to control your analyzer using internal IBASIC or with an external computer connected via GPIB. The analyzer's internal CPU running IBASIC programs acts as a complete system controller residing inside your analyzer. You have the additional capability of controlling the analyzer over a LAN. Refer to the *LAN Interface User's Guide Supplement* (supplied with the analyzer) for information.Some of the analyzer features that support automation are built-in and operate solely on the analyzer; limit testing is one example of a built-in automation feature. Other features can be executed by programs running internally under IBASIC or on an external computer.

NOTE

Example programs can be found in the following four locations:
- Web site **http://www.agilent.com**. Use the search function to find Web pages related to 8712 example programs.
- Example Programs Disk, 8712ET/ES and 8714ET/ES (DOS format): part number 08714-10003
- Example Programs Disk, 8712ET/ES and 8714ET/ES (LIF format): part number 08714-10004
- *Example Programs Guide, 8712ET/ES and 8714ET/ES*: part number 08714-90016

Measurement System Configurations and Considerations

When configuring your test system, there are many things to consider, such as:

- how many test stations do you need now?
- how many test stations will be needed in the future?
- how much space is available at each test station?
- what type of testing will be done?
- how will the measurements be controlled?
- how will the data be analyzed and archived?
- what level of throughput is required?

After answering these questions, you should decide which of the following configurations best meets your needs.

Stand-Alone Network Analyzers

In the stand-alone configuration, you can either take advantage of the operator-initiated automation features, or you can use the internal CPU and IBASIC programs to further automate your measurements.

Stand-Alone Analyzer under Operator Control

In this configuration, the measurement is controlled directly by the operator, with very little automation. No computer or IBASIC control is used, however the fast-recall feature may be used to quickly change between different instrument states. This configuration is well suited for simple pass/fail testing using the built-in limit-testing features. Configure your system as a stand-alone analyzer if you would like to

- simplify test-system configuration
- maintain full control by operators
- minimize test-development time

Stand-Alone Analyzer under IBASIC Control

In this configuration, the measurement is controlled by an IBASIC program running inside the analyzer. With IBASIC, the measurement setup and control can be highly automated, reducing the burden on the operator. Since the measurement is under programmatic control, data can be collected and analyzed in order to monitor your process and quality. IBASIC's keystroke recording lets you construct programs quickly, without needing to refer to the programming documentation. Using AUTOST files, the analyzer will load and run your program when power is turned on.

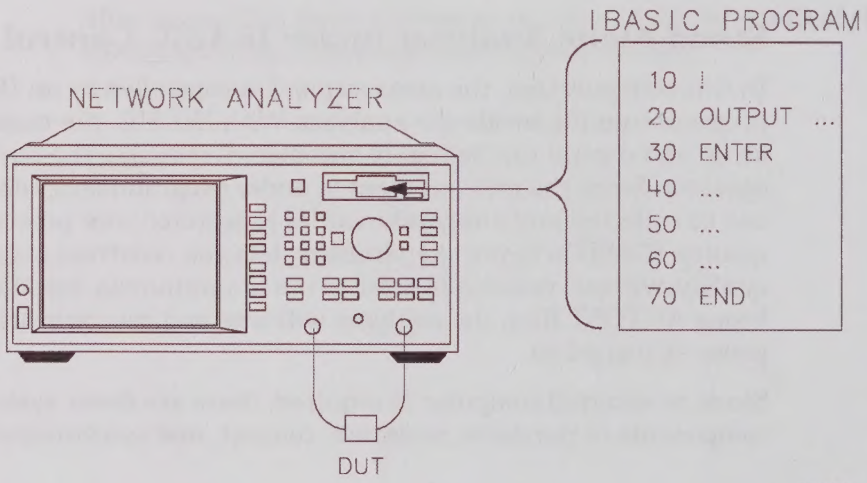
Since no external computer is required, there are fewer system components to purchase, maintain, connect, and synchronize.

Configure your system as a stand-alone instrument with IBASIC if you would like to

- simplify test-system configuration
- minimize the space required for a system
- simplify programming with keystroke recording
- use (BEGIN) key macros
- automate measurement setup and control
- simplify measurements
- collect data
- run application programs on the analyzer

Figure 2-1 depicts a stand-alone network analyzer running IBASIC.

Figure 2-1 Stand-Alone Network Analyzer Running IBASIC



External Computer Control

You can use an external computer to control your analyzer or test system by using the LAN or GPIB interface on your analyzer. Additionally, you can use an external computer in conjunction with the analyzer's internal CPU running IBASIC for optimum throughput. (For more information on using the LAN interface, refer to *The LAN Interface User's Guide Supplement*.)

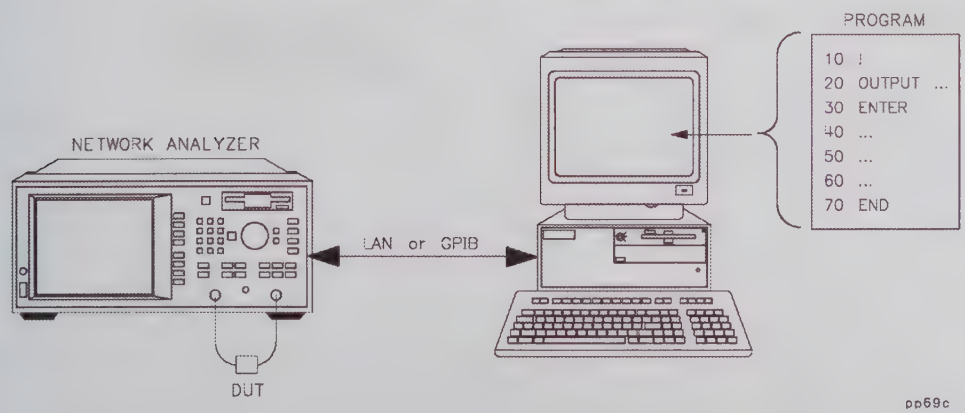
External Computer Alone

In this configuration, the measurement is controlled by a computer external to the analyzer, using the LAN or the GPIB interface. The measurement setup and control can be highly automated and designed for optimum throughput. Data can be automatically collected and analyzed in order to monitor quality and processes. You can connect one or more analyzers to each computer. Since the computer can be connected to other computers via a Local Area Network (LAN), measurement statistics can be easily tracked and archived using computer applications.

Configure your system as an analyzer controlled by an external computer if you would like to

- centralize automation and application programs
- develop a more sophisticated data collection system
- add networking capability

Figure 2-2 **Network Analyzer Controlled by a Computer**

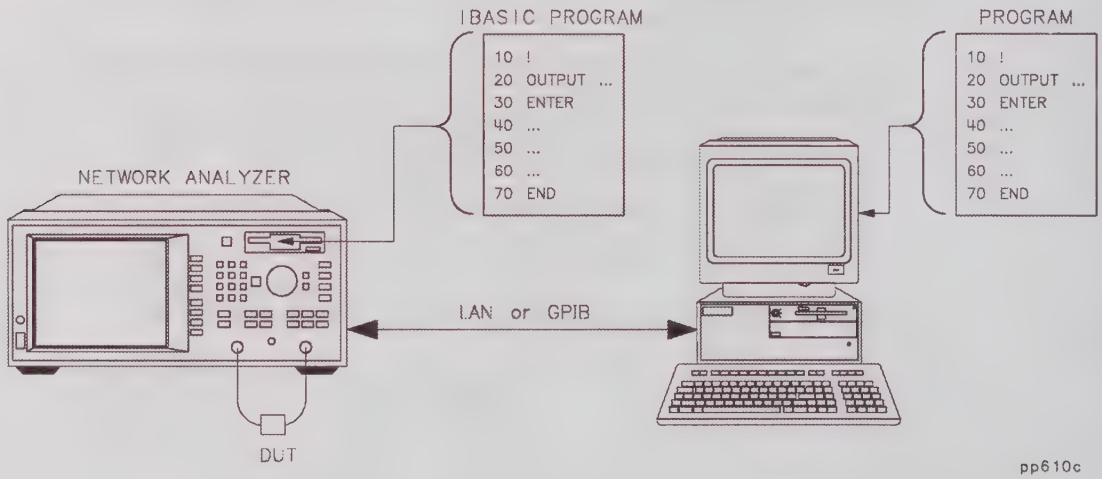


External Computer in Conjunction with IBASIC

In this configuration, the measurement is controlled by an IBASIC program running inside the analyzer. IBASIC can provide high-speed measurement control and data collection, and save the results in program memory or on a disk. The external computer then communicates with IBASIC, and collects the measurement results at some defined interval. This configuration can result in higher throughput, especially if the measurement setup and control is complex. Configure your system as an analyzer with IBASIC and an external computer if you would like to

- centralize automation and application programs
- develop a more sophisticated data collection system
- add networking capability
- add local-automation capability

Figure 2-3 **Network Analyzer Running IBASIC, Controlled by a Computer**



Expandability and Large Systems

You can connect up to 10 analyzers to a single computer without compromising maximum GPIB bus speed. The IEEE-488 standard states that the bus can achieve a data rate of 500 KBytes per second for buses up to 20 meters in length, with up to 10 devices (one device per 2 meters of cable).

IEEE-488 provides for 31 unique addresses (0 through 30), however it restricts the number of devices on the bus to 15. Due to this restriction, you may need to add more computers as you add more analyzers. Set each analyzer's address via the (SYSTEM OPTIONS) **GPIB** menu.

With your analyzer's LAN feature, you can overcome the physical limitations of GPIB cabling and create large systems with hundreds of analyzers. 10-BaseT LAN provides a theoretical bandwidth of close to 1 MByte/second, although in practice, throughput will be slower. See *The LAN Interface User's Guide Supplement* for more information.

Throughput Considerations

When considering the throughput of the system configuration, contributing factors are

- device handling time
- operator interaction time
- measurement speed
- data transfer speed
- computation speed (when applicable)

For more information, refer to *Improving Throughput in Network Analyzer Applications*, application note number AN 1287-5 (or literature number 5966-3317E).

Selecting a Measurement Controller

There are three standard configurations that you can use to control the analyzer:

- stand-alone operation using IBASIC
- computer-controlled GPIB operation
- computer-controlled LAN operation

Stand-Alone Operation using IBASIC

IBASIC, in effect, puts a controller inside your analyzer and eliminates the need for an external computer. IBASIC controls the analyzer by sending SCPI commands to address 800 (`OUTPUT 800;"Command"`), or by using high-speed built-in subprograms. Since IBASIC shares CPU time with the analyzer, it may cause some degradation in measurement throughput if your program performs intensive computations. However, for most applications, it provides excellent performance and convenience. Refer to the *HP Instrument BASIC User's Handbook* and *HP Instrument BASIC User's Handbook Supplement* for more information.

Computer-Controlled GPIB Operation

An external computer can be used to control the analyzer. It can be a personal computer (PC) or a dedicated HP BASIC controller. The external computer sends standard SCPI commands to address 716 (default) (`OUTPUT 716;"Command"`) to control the analyzer. Refer to the *Programmer's Guide* for more information.

You can use one computer to control several analyzers (see "Expandability and Large Systems" earlier in this chapter). However, if a large number of SCPI commands are required per measurement, throughput may be degraded. Typical limits are 3 to 10 analyzers per computer.

Computer-Controlled LAN Operation

An external computer can send SCPI commands via the LAN using a telnet connection. Refer to *The LAN Interface User's Guide Supplement* for more information.

You can use one computer to control several analyzers. Performance will typically be limited by the computer's speed, which in turn will limit the number of analyzers per computer.

Using IBASIC and External Control Together

The analyzer can be controlled by commands from an external computer, internal IBASIC, or from both at the same time, as long as certain precautions are observed. Things to consider:

1. If both the analyzer and the computer send SCPI commands at the same time, the analyzer may not finish the IBASIC command before executing the computer's command, or vice-versa. The programmer must ensure that SCPI commands executed by IBASIC do not overlap with SCPI commands sent from an external computer, otherwise the system may lock up. Synchronization between the analyzer and the external computer must be ensured. See "Synchronizing IBASIC with an External Controller" in *HP Instrument BASIC User's Handbook Supplement* for additional information.
2. Both IBASIC (SelectCode 8) and the external computer (LAN or SelectCode 7) share the same GPIB status model (the same analyzer status bits go to each). Be careful sending commands which affect status reporting, such as `*CLS`, `STAT:PRES`, `*RST`, etc.

Selecting a Programming Language

HP BASIC

HP BASIC has long been a favorite programming language for instrument control. It features an extensive list of keywords, and powerful `OUTPUT` and `ENTER` formatting, making it easy to perform common tasks. This generally results in very high programming productivity. HP BASIC runs on HP series 700 workstations and series 300 instrument controllers.

IBASIC

IBASIC is a version of BASIC that runs inside of the network analyzer.

IBASIC is a sub-set of HP's BASIC. It has roughly the same keywords as HP BASIC 4.0. With very little effort, you can design your program so that it will run either inside the analyzer or on a computer with no modification.

The IBASIC program runs concurrently with normal instrument measurement processing. Since IBASIC has direct access to the analyzer's measurement arrays, it can read them and write to them very quickly, eliminating the need to use SCPI commands. Using IBASIC's keystroke recording, you can write a large portion of your instrument control program by pressing the keys on the analyzer's front panel. IBASIC can be used in a stand-alone instrument, or in conjunction with an external computer.

For more information on IBASIC, refer to the *HP Instrument BASIC User's Handbook* and *HP Instrument BASIC User's Handbook Supplement* that were shipped with your analyzer.

HP VEE

HP VEE is a powerful application which lets you graphically create programs to control your instrument. VEE automatically handles the programming details so you can focus on higher level tasks. It also contains statistical functions which you can use to monitor your production process.

VEE runs on PCs as well as HP-UX and Sun Workstations.

VEE can be used in conjunction with a VEE instrument driver. The VEE instrument driver presents a picture of the instrument's front panel on the computer display. Using the mouse, you can click on the front panel keys to control the instrument, similar to IBASIC keystroke recording.

For information on VEE, including literature and preview disks, please call the nearest Agilent instrument support center, or sales or service office. Refer to the front of this book for a table of sales and service offices.

Microsoft® QuickBasic™

QuickBasic has been a popular programming language since it runs on PCs. It does *not* offer a rich keyword set as does HP BASIC, and is *not* optimized for instrument control. To control the analyzer via GPIB, an GPIB card and driver library must be installed in the computer. The driver library will provide subroutines such as `IOOUTPUT` and `IOENTER` which let you control your analyzer.

C and C++

If you are using C or C++, you will need a driver library to use your GPIB card. Agilent offers a library called Standard Instrument Control Library (SICL). SICL is available for PCs running Microsoft Windows® and using HP's GPIB card. SICL is also available on HP series 700 UNIX workstations.

For LAN communication with your analyzer, C or C++ is often used. Multi-threaded programs can be created to allow easy and precise control of many analyzers operating asynchronously. This approach maximizes speed and throughput.

VXI *plug&play* Driver

Your analyzer is VXI *plug&play* compliant. VXI *plug&play* drivers provide a high level control interface for the analyzer, thereby reducing the level of instrument-specific knowledge required to program the analyzer. The driver for the 8712ET/ES and 8714 ET/ES functions in Win95, Win98, and WinNT (3.51 or higher) and was designed for the following programming platforms:

- Microsoft Visual C++ version 4.0 or higher
- Microsoft Visual BASIC version 5.0 or higher
- Borland C++ version 4.5 or higher
- HP VEE version 3.2 or higher
- National Instruments Lab Window/CVI version 4.0.1 or higher
- National Instrument LabVIEW version 5.00 or higher

The free VXI *plug&play* driver for your analyzer should be available soon on the Web at:

http://www.tm.agilent.com/tmo/software/English/comptest_VXI_drivers.html
OR
http://www.agilent.com/ena

Example Programs

The *Programmer's Guide* contains detailed information on controlling the analyzer via GPIB. The *Example Programs Guide* includes several example programs written in HP BASIC and in IBASIC.

The *HP Instrument BASIC User's Handbook* and *HP Instrument BASIC User's Handbook Supplement* contain detailed information on how to control the analyzer using IBASIC.

The LAN Interface User's Guide Supplement contains detailed information on controlling the analyzer via LAN, including example programs.

3 Operator Interaction

Many tests are performed by technicians, operators, or testers who interact with the measurement system. When designing the automation system, it is important that the system allow operators to perform the measurement tasks quickly and consistently. The system must also be easy to learn and easy to use, providing the user with instructions and feedback.

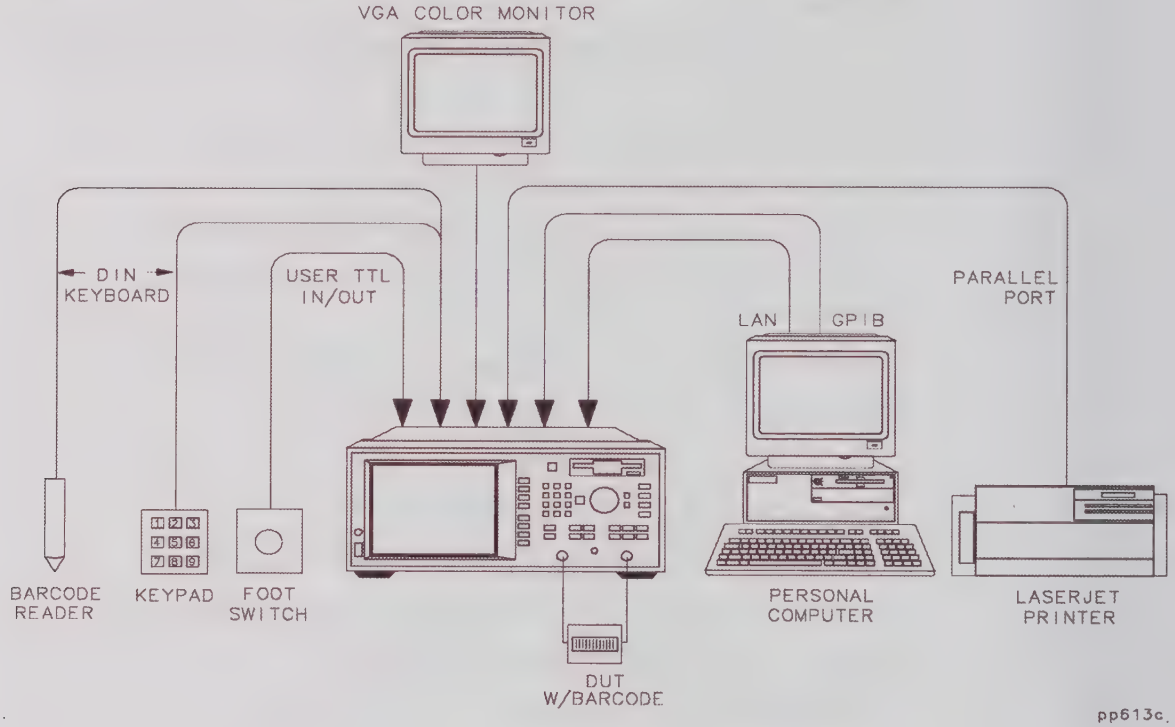
The analyzer provides many features to satisfy these requirements. The features include

- user-defined pop-up messages to prompt the operator
- on-screen graphics to create custom diagrams
- user-defined measurement channel and frequency annotation
- ability to define the (BEGIN) key menu with frequently-used custom softkeys
- IBASIC display window; configurable as full or split
- IBASIC `DISP` line and `INPUT` line
- data entry using a barcode reader
- data entry using an external keyboard
- hot keys on external keyboard for common functions
- operator control of measurements using a foot or hand switch
- beeper with adjustable volume and pitch
- limit test pass/fail TTL output
- user-defined TTL input/output
- output for external VGA-compatible color monitor

The following sections explain how to use these features, and show several examples.

Figure 3-1 shows an example test system setup which utilizes the extensive connectivity capabilities of the analyzer. All of the interface ports shown are standard equipment on your analyzer.

Figure 3-1 Example Test System Setup



Prompting the Operator

You can display a customized message in the center of the analyzer's display by using the following SCPI command:

```
DISPlay:ANNotation:MESSage
```

For example,

```
OUTPUT @Rfna;"DISP:ANN:MESS 'Connect device, then press button'"
```

You can specify how long you want the message to remain on the screen by using one of the following timeout words: `SHORt`, `MEDium`, `LONG`, `NONE`. For example,

```
OUTPUT @Rfna;"DISP:ANN:MESS 'Test passed.'",MEDIUM
```

To clear the message immediately, use the following command:

```
DISPlay:ANNotation:MESSage:CLEar
```

The message string can contain a maximum of 25 lines with up to 47 characters per line. However, it cannot be more than 254 characters in length, including carriage returns and line feeds.

If you are using IBASIC, you can use the BASIC keyword `DISP` to display a small one-line message near the bottom of the screen. For example,

```
DISP "Connect device to REFLECTION port"
```

Using Graphics to Create On-Screen Diagrams

Diagrams showing how to connect devices or perform measurements can be a powerful tool. You can draw customized diagrams on the analyzer's display using the following SCPI commands:

```
DISPlay:WINDow[1|2|10]:GRAPhics:command
```

These commands let you draw lines, rectangles, circles, and text onto the screen. The number specified in the WINDow part of the command selects where the graphics are to be drawn:

WINDow1 draws the graphics to the measurement channel 1 window.

WINDow2 draws the graphics to the measurement channel 2 window.

WINDow10 draws the graphics to the IBASIC display window.

Using split display, you can display the measurement in one half of the screen while displaying a connection diagram in the other half.

For more details on SCPI graphics commands, refer to "Using Graphics" in the *Programmer's Guide*.

If you are using IBASIC, you can use the BASIC graphics keywords such as MOVE and DRAW to draw diagrams in the IBASIC window.

```
MOVE X1,Y1
DRAW X2,Y2
```

Since IBASIC keywords always draw to the IBASIC window, you cannot use them to draw in the same window as your measurement. To draw in the measurement window, your IBASIC program must use the standard SCPI commands.

```
DISP:WIND1:GRAPHICS:command
```

and

```
DISP:WIND2:GRAPHICS:command
```

For more details on IBASIC graphics, refer to "Graphics and Display Techniques" in the *HP Instrument BASIC User's Handbook Supplement*.

User-Defined BEGIN Key Menu

The **User BEGIN** feature adds the following capabilities:

- Create softkeys to allow one-key-press operation of functions that usually take several key presses.
- Reassign softkeys to implement your most-used functions/features.
- Create softkeys to implement new features created with IBASIC—for example, a gain-compression measurement.
- Create softkeys to implement application support.

This feature is designed to provide the fastest possible sweep speeds while taking advantage of the flexibility provided by IBASIC. This is the simplest way for recalling instrument states or configuring most-used softkey functions under a single softkey menu. The eighth softkey in the (BEGIN) menu provides access to the menus below. Toggle the eighth softkey to enable/disable (BEGIN) or **User BEGIN** . You cannot redefine the eighth softkey.

(BEGIN)		**User BEGIN**
Amplifier		User label 1
Filter		User label 2
Broadband Passive		User label 3
Mixer		User label 4
Cable (Option 100 only)		User label 5
		User label 6
		User label 7
User BEGIN on OFF	——>	User BEGIN ON off

Once you have selected a menu, the same menu will be displayed for subsequent key presses of the (BEGIN) hardkey.

By selecting (SYSTEM OPTIONS) **System Config** **User TTL Config** **Softkey Auto-Step** , you can automatically step through the user-defined (BEGIN) menu keys by use of a switch connected to the USER TTL IN/OUT rear panel connector. See "Using a Switch with User BEGIN" on page 3-11 for more information.

Selecting the **User BEGIN** softkey will run a macro function defined by a sequence of IBASIC commands defined within an IBASIC program. IBASIC programs to be used for **User BEGIN** must have the following structure:

```
10      !The following label must be present.  DO NOT REMOVE
20      User_begin: !
30      ...
40      ! Define softkey labels
50      OUTPUT @871X;"DISP:MENU2:KEY1 'Test Setup 1';*WAI"
60      OUTPUT @871X;"DISP:MENU2:KEY2 'Test Setup 2';*WAI"
70      OUTPUT @871X;"DISP:MENU2:KEY3 'Save Results';*WAI"
80      OUTPUT @871X;"DISP:MENU2:KEY4 'Print Results 4';*WAI"
100     User_pause: PAUSE
110     GOTO User_pause
120     !The following key labels must be present.  DO NOT REMOVE
140     User_key1: !Insert code for softkey 1 here
150     GOTO User_pause
160     User_key2: !Insert code for softkey 2 here
170     GOTO User_pause
        ...
200     User_key7: !Insert code for softkey 7 here
210     GOTO User_pause
```

The labels required are:

User_begin

User_pause

User_key1

User_key2

User_key3

User_key4

User_key5

User_key6

User_key7

NOTE Your **User BEGIN** program must contain all of these labels, even if you are not using all of the softkeys.

A default **User BEGIN** program is created automatically when there is no IBASIC program installed. In the default program, the third softkey is defined to be the **Mkr --> Max** function, the fourth softkey prompts the user for a title, and also enables the clock. The default program is listed next. You may edit this program to change the functions you need. Once you have edited the program, be sure to save the program to memory for later recall.

```
1     ! The following line is required. DO NOT REMOVE!
2 User_begin:     ASSIGN @Hp8714 TO 800 ![User Begin] Program
3     !
4     ! To Modify:
5     ! Use [IBASIC][EDIT] or [IBASIC][Key Record]
6     !
7     !
8     ! Declare storage for variables.
9     DIM Name$[60],Str1$[60],Str2$[60],Str3$[60]
10    !
11    ! Clear the softkey labels
12    OUTPUT @Hp8714;"DISP:MENU2:KEY8 '';*WAI"
13    !
14    ! Re-define softkey labels here.
15    OUTPUT @Hp8714;"DISP:MENU2:KEY1 '*';*WAI"
16    OUTPUT @Hp8714;"DISP:MENU2:KEY2 '*';*WAI"
17    OUTPUT @Hp8714;"DISP:MENU2:KEY3 'Mkr -> Max';*WAI"
18    OUTPUT @Hp8714;"DISP:MENU2:KEY4 'Title and Clock';*WAI"
19    OUTPUT @Hp8714;"DISP:MENU2:KEY5 '*';*WAI"
20    OUTPUT @Hp8714;"DISP:MENU2:KEY6 '*';*WAI"
21    OUTPUT @Hp8714;"DISP:MENU2:KEY7 '*';*WAI"
22    !
23    !The following 2 lines are required. DO NOT REMOVE!
```

```
24 User_pause:   PAUSE
25    GOTO User_pause
26    !
27 User_key1:        ! Define softkey 1 here.
28    GOSUB Message ! Remove this line.
29    GOTO User_pause
30    !
31 User_key2:        ! Define softkey 2 here.
32    GOSUB Message ! Remove this line
33    GOTO User_pause
34    !
35 User_key3:        ! Example Marker Function
36    OUTPUT @Hp8714;"CALC1:MARK1 ON"
37    OUTPUT @Hp8714;"CALC1:MARK:FUNC MAX"
38    GOTO User_pause
39    !
40 User_key4:        ! Example Title Entry
41    INPUT "Enter Title Line 1.  Press [Enter] when done.",Name$
42    OUTPUT @Hp8714;"DISP:ANN:TITL1:DATA '""'"
43    OUTPUT @Hp8714;"DISP:ANN:TITL ON"
44    GOTO User_pause
45    !
46 User_key5:        ! Define softkey 5 here.
47    GOSUB Message ! Remove this line.
48    GOTO User_pause
49    !
50 User_key6:        ! Define softkey 6 here.
51    GOSUB Message ! Remove this line.
52    GOTO User_pause
53    !
54 User_key7:        ! Define softkey 7 here.
55    GOSUB Message ! Remove this line.
56    GOTO User_pause
57    !
58 Message:    !
59    Str1$="This key is programmable."
60    Str2$="To modify, select"
61    Str3$="[System Options], [IBASIC], [Edit]."
62    OUTPUT @Hp8714;"DISP:ANN:MESS '""', MEDIUM"
63    RETURN
64    !
65    END
```

NOTE

For more **User BEGIN** programs, see the *Example Programs Disk* and the *Example Programs Guide.*

Loading a User BEGIN Program

A **User BEGIN** program can be automatically loaded at power up if the program is named "AUTOST". An "AUTOST" program is loaded at power up from the internal non-volatile memory or from a 3.5 inch floppy disk inserted into the analyzer's disk drive. (To manually load and run an "AUTOST" file, press (BEGIN) **Autost** .) A **User BEGIN** program does not have to be an "AUTOST" program; it can also be loaded as any other IBASIC program. (See the *HP Instrument BASIC User's Handbook Supplement* for more information.)

When the **User BEGIN** key is pressed, the program will remain idle until it is needed. The program remains idle until a softkey is pressed and code related to that softkey is executed. After the code is executed, the program returns to idle. Use of **User BEGIN** does *not* restrict access to any normally available front panel feature, nor does this key affect sweep update rates.

Modifying a User BEGIN Program

You can modify the User BEGIN program with the built-in editor, an ASCII file editor on a computer, or with keystroke recording. For example, to modify the default program to recall a setup:

1. Select (SYSTEM OPTIONS) **IBASIC** **Edit** . Move the edit cursor to line:

   ```
   15    OUTPUT @Hp8712;"DISP:MENU2:KEY1 '*';*WAI"
   ```

2. Use an external keyboard to replace `'*'` with `'Setup 1'`.
3. Move the edit cursor to line 28. Delete the line.
4. Use keystroke recording to create a setup function if you like, or you can now insert code you have written.

To use keystroke recording to modify the program:

5. Exit the editor by selecting **Prior Menu** .
6. Enable keystroke recording with **Key Record ON** .
7. Now perform the keystrokes required for Setup 1.
8. When the setup is completed, select (SYSTEM OPTIONS) **IBASIC** **Key Record OFF** .

9. To verify your change, select (PRESET) (BEGIN) **User BEGIN ON** .
10. Select the first softkey which should be labeled "Setup 1". This should return you to your correct setup.
11. You may save this program as an AUTOST file or other file for later recall.

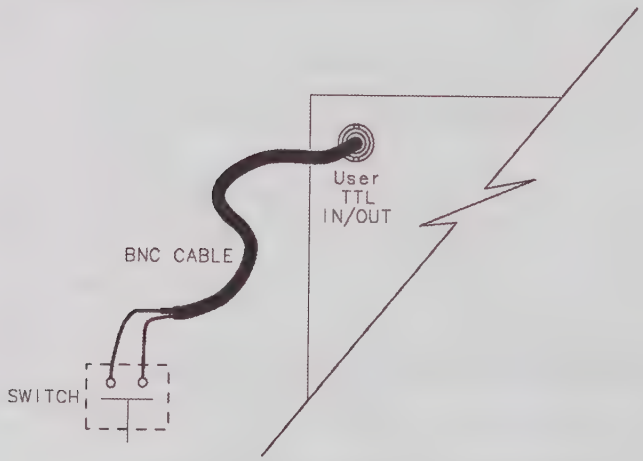
Refer to the *HP Instrument BASIC User's Handbook Supplement.*

Using a Switch with User BEGIN

When user-defined (BEGIN) is used in conjunction with a switch connected to the USER TTL IN/OUT rear-panel connector, you can cycle through up to seven softkeys in sequence by activating the switch.

1. Connect a switch to the USER TTL IN/OUT rear panel connector as shown in Figure 3-2.

Figure 3-2 Connecting a Switch to the USER TTL IN/OUT Connector



po699b

2. Make sure the analyzer is configured to use the USER TTL IN/OUT connector for softkey sequencing: press (SYSTEM OPTIONS) **System Config** **User TTL Config** **Softkey Auto-Step** .
3. Press the switch several times while observing the analyzer.
4. Notice that with each press of the switch, the softkey labels are highlighted (boxed) in succession, and that after the last available key has been used, the sequence starts again at the top of the softkey menu.

Using a Barcode Reader for Data Entry

Devices under test (DUTs) are often labeled with a serial number and part number. If the DUT is labeled with this information in barcode format, a barcode reader can be used to enter the DUT information into the analyzer or into the computer controlling the analyzer. Doing so provides a simple and safe link between the DUT and the measurement data. Information such as the operator's name or test station number can also be entered, to allow correlation of the devices tested with the test station.

Connect a barcode reader, such as the Agilent KeyWand HBKW-1220, to the analyzer's DIN KEYBOARD connector (on the rear panel). Once connected, the barcode reader will send scanned barcode characters to the analyzer just as if they were typed on a keyboard. The barcode characters will be followed by a carriage return. The barcode wand and the external keyboard can be connected simultaneously.

See "Using the Analyzer's Title Feature" on page 3-15 for more information on using the title feature with a barcode reader.

In addition to Agilent's barcode wand, other vendors offer products such as cordless barcode readers, laser scanner readers, barcode label printers, barcode fonts for Windows®, and barcode labeling software. For more information about Agilent Technologies barcode products—including listings for distributors worldwide—please visit our Web page at http://www.semiconductor.agilent.com/barcode/.

Using an External Keyboard for Data Entry

An IBM PC-AT compatible keyboard can be connected to your analyzer's DIN KEYBOARD connector and used to quickly and conveniently enter file names for instrument state save/recall, or text for title lines.

NOTE

If your keyboard has a standard (large) DIN connector, you will need to use a DIN to mini-DIN adapter to connect the keyboard to the analyzer. These adapters are available as part number 1252-4141. Contact the nearest Agilent sales or service office for more information.

See "Using the Analyzer's Title Feature" on page 3-15 for more information.

Using the Analyzer's Title Feature

The analyzer has two 30-character title lines which can be entered using the barcode reader. From the front panel, press (DISPLAY) **More Display Title and Clock Title Line 1** , *or* on an external keyboard, press F9. Then use the barcode reader to scan in the information from the DUT or use a keyboard to type in the information. Once stored in the title line, the information will be included on hardcopy dumps. The title lines can also be set or queried using the following SCPI command:

```
DISPlay:ANNotation:TITLe[1|2]:DATA
```

For example:

```
OUTPUT @Rfna;"DISP:ANN:TITL1:DATA 'BPF-177, SN US95170001'"
```

and

```
OUTPUT @Rfna;"DISPlay:ANNotation:TITLe1:DATA?"
ENTER @Rfna;Title1$
```

Use the command

```
"DISPlay:ANNotation:TITLe[1|2] ON|OFF"
```

to display or hide the title. If you are using IBASIC, you can use the INPUT statement to read in barcode or keyboard characters. For example:

```
30  INPUT "Scan in the Barcode now",Dut$
40  OUTPUT 800;"DISP:ANN:TITL1:DATA '";Dut$;"'"
50  OUTPUT 800;"DISP:ANN:TITL1 ON"
60  END
```

Hot Keys on External Keyboard for Common Functions

You can use a keyboard's keys instead of the analyzer's keys to control the analyzer. It provides the same information that can be found on a template that is supplied with each analyzer (part number 08712-80028). Function keys F9, F10, and F11 are "hot keys" which perform common operations such as entering measurement titles and saving measurement results to disk.

Table 3-1 Keyboard Template Definition

Keyboard Key Name	Analyzer Function	Keyboard Key Name	Analyzer Function
F1	Softkey 1	Shift-F1	MEAS 1
F2	Softkey 2	Shift-F2	MEAS 2
F3	Softkey 3	Shift-F3	FREQ
F4	Softkey 4	Shift-F4	POWER
F5	Softkey 5	Shift-F5	SWEEP
F6	Softkey 6	Shift-F6	MENU
F7	Softkey 7	Shift-F7	SCALE
F8	Softkey 8	Shift-F8	MARKER
F9	Title Keys Line 1 *or* RCL[1]	Shift-F9	DISPLAY
F10	Title Keys Line 2 *or* Edit[1]	Shift-F10	FORMAT
F11	Title Keys Clock *or* Window[1]	Shift-F11	CAL
F12	Re-save File *or* Run[1]	Shift-F12	AVG
Esc	IBASIC Command Line On/Off	Ctrl-F2	HARDCOPY
Print Screen	Help	Ctrl-F3	SYSTEM OPTIONS
Ctrl-Print Screen	Hardcopy Keyboard Template	Ctrl-F4	PRESET
Shift-Print Screen	Hard Copy Graph and Softkeys	Ctrl-F5	BEGIN
Ctrl-F1	SAVE RECALL		

1. When the IBASIC command line is enabled

User-Defined TTL Input/Output

The USER TTL port can be used as a general-purpose input or output. Like the LIMIT TEST IN/OUT line, the USER TTL IN/OUT line is an open-collector drive. When used as an input, the state of the USER TTL IN/OUT line can be read with either the SCPI command

```
"DIAG:PORT:READ? 15,1"
```

or with the IBASIC command

```
I = READIO(15,1)
```

When used as an output, the state of the USER TTL IN/OUT line can be set with either the SCPI command

```
"DIAG:PORT:WRITE 15,1,value"
```

or with the IBASIC command

```
WRITEIO 15,1;value
```

The USER TTL IN/OUT port can also be used in conjunction with an external switch for softkey auto-stepping, or as a "sweep out" port. Be sure that USER TTL IN/OUT port is configured properly for general-purpose I/O by pressing (SYSTEM OPTIONS) **System Config** **User TTL Config** **Default** . Or use the following SCPI command:

```
SYST:COMM:TTL:USER:FEED DEFAULT
```

CAUTION Be sure to observe static precautions when using this port.

Using a Foot or Hand Switch

You can connect a foot switch, button box, or custom keyboard which has a few function keys that are custom-labeled, and use this in conjunction with IBASIC to allow consistent, error-free step-by-step measurement control. The operator presses one key, then the next, in order.

NOTE

A switch can also be used with the analyzer's fast-recall feature. See "Using Fast Recall with a Switch" on page 4-5.

The foot-switch simply connects two wires together, grounding the center pin of the analyzer's USER TTL IN/OUT rear panel connector. (See Figure 3-2.) The status of the USER TTL IN/OUT can be read using the SCPI commands:

```
30  OUTPUT 716;"DIAG:PORT:READ? 15,1"

40  ENTER 716;X
```

or the IBASIC command:

```
30  X=READIO(15,1)
```

When the foot-switch is open, the variable "X" will be set to 1. When it is closed, the variable "X" will be set to 0. Switch debounce is generally not a problem, due to the relatively slow polling rate of the program.

Refer to "Analyzer Port Numbers" on page 3-22 for tables describing the various analyzer ports that you can access using SCPI or IBASIC commands.

Following is an example program which shows how to display a message and read the foot switch to control your measurements. This program is named "TTL_IO" on your *Example Programs Disk*.

For an example which uses the IBASIC `READIO` command, refer to the program "USER_BIT" on your *Example Programs Disk*.

```
100   ! Filename: TTL_IO
110   !
120   ! This program reads the USER TTL IO
130   ! port, and counts how many times a
140   ! switch connected to the port is pressed.
150   !
160   DIM Msg$[200]
170   INTEGER X
180   !
190   IF POS(SYSTEM$("SYSTEM ID"),"HP 871") THEN
200     ASSIGN @Hp8711 TO 800
```

```
210   ELSE
220     ASSIGN @Hp8711 TO 716
230     ABORT 7
240     CLEAR 716
250   END IF
260   !
270   Pass_count=0
280   Start: !
290   LOOP
300      ! Display message
310      Msg$="'DUTs passed: "&VAL$(Pass_count)&CHR$(10)
320      Msg$=Msg$&"Press button to measure next DUT.'"
330      OUTPUT @Hp8711;"DISP:ANN:MESS ";Msg$
340      !
350      ! Wait for button to be pressed
360      REPEAT
370        OUTPUT @Hp8711;"DIAG:PORT:READ? 15,1"
380        ENTER @Hp8711;X
390      UNTIL X = 0
400      DISP "Button is now pressed."
410      OUTPUT @Hp8711;"DISP:ANN:MESS:CLEAR"
420      !
430      ! Wait for button to be released
440      REPEAT
450        OUTPUT @Hp8711;"DIAG:PORT:READ? 15,1"
460        ENTER @Hp8711;X
470      UNTIL X = 1
480      DISP "Button is now released."
490      !
500      OUTPUT @Hp8711;"DISP:ANN:MESS 'Measuring...'"
510      ! Add code here to take sweep
520      ! and measure DUT.
530      WAIT 1
540      Pass_count=Pass_count+1
550   END LOOP
560   END
```

Limit-Test Pass/Fail TTL Input/Output

When limit testing is turned on, the LIMIT TEST IN/OUT rear-panel BNC connector indicates the status of the limit test. If the limit test passes, this TTL output goes high. If the limit test fails, this TTL output goes low. This signal can be used, for example, as an input to a materials handler.

CAUTION

Since the TTL output has limited current drive capability, it should be buffered when controlling high current devices such as mechanical relays. Otherwise, damage to the instrument may result.

The limit-test TTL can also be used as a general-purpose input, since the analyzer drive to this line is open collector. When used as an input, limit testing should be turned *off* so the instrument will allow the limit test line to float high. The line can then be connected to an external switch which should only pull the signal to ground or let it float (an external circuit should not drive this line). A push button or foot switch can be attached to the line to pull the signal to ground.

The state of the signal can be monitored by the automated system to determine when the operator is ready for some action. The state of the limit TTL line can be read with either the SCPI commands

```
20  INTEGER X
30  OUTPUT 716;"DIAG:PORT:READ? 15,2"
40  ENTER 716;X
```

or using the IBASIC command

```
30  X=READIO(15,2)
```

Finally, the LIMIT TEST TTL IN/OUT line can be used as a general-purpose output line. With limit testing turned off, the state of the line can be set to logic high or low with either the SCPI command

```
"DIAG:PORT:WRITE "
```

or with the IBASIC command

```
"WRITEIO ,value".
```

The following is an example of reading the LIMIT TEST TTL IN/OUT line when used as an input:

```
30  Limit = READIO(15,2)
40  ! The "Limit" variable will be set to 0 if the signal is low
50  ! and 1 if the signal is high.
```

Analyzer Port Numbers

Table 3-2 Writeable Ports

Port Number	Register	Description
15	0	Outputs 8-bit data to the Cent_D0 thru D7 lines of the Centronics port. Cent_D0 is the least significant bit, Cent_D7 is the most significant bit. Checks Centronics status lines for: Out of Paper Printer Not on Line BUSY ACKNOWLEDGE
15	1	Sets/clears the user bit according to the least significant bit of A. A least significant bit equal to 1 sets the user bit high. A least significant bit of 0 clears the user bit.
15	2	Sets/clears the limit pass/fail bit according to the least significant bit of A. A least significant bit equal to 1 sets the pass/fail bit high. A least significant bit of 0 clears the pass/fail bit.
15	3	Outputs 8-bit data to the Cent_D0 thru D7 lines of the Centronics port. Cent_D0 is the least significant bit, Cent_D7 is the most significant bit. Sets the Printer_select signal high (de-select). Does not check Centronics status lines.
9	0	Outputs a byte to the serial port. The byte is output serially according to the configuration for the serial port.

NOTE When using the `WRITEIO(15,0)` or `WRITEIO(15,3)` command, the Printer_Select Line is set high. However, when the instrument is doing hardcopy, the Printer_Select Line is set low. The Printer_Select line may or may not be used by individual printers. Check with your printer manual.

Table 3-3 Readable Ports

Port Number	Register	Description
9	0	Reads the serial port.
15	0	Reads the 8-bit data port Cent_D0 thru D7.
15	1	Reads the user bit.
15	2	Reads the limit test pass/fail bit.
15	10	Reads the 8-bit status port. D0 — Cent_acknowledge D1 — Cent_busy D2 — Cent_out_of_paper D3 — Cent_on_line D4 — Cent_printer_err

Output for External VGA-Compatible Monitor

You can connect a VGA-compatible external monitor to the VIDEO OUT COLOR VGA connector for a large-screen color view of your measurement if you wish. See "Using an External VGA Monitor" in Chapter 4 of your analyzer's *User's Guide* for information on using an external monitor with your system.

4 Measurement Setup and Control with Fast Recall

The production and testing of RF components often involves several steps, with each step requiring a unique set of instrument settings. Likewise, the different test configurations at each step may require associated calibrations. Manually entering these sets of parameters (or "states") or calibrating at each step in the manufacturing process is slow, prone to operator error, and costly. The fast-recall feature allows you to recall one of seven instrument states with just a single key press, *or* to cycle through up to seven different instrument states with a switch.

Using Fast Recall with the Front Panel or a Keyboard

NOTE

The following explanation assumes that you are familiar with the information presented in "Saving and Recalling Measurement Results" in Chapter 4 of your analyzer's *User's Guide*.

1. Press (SAVE RECALL).
2. If the measurement display area changes to a listing of files on the currently selected disk, fast recall is off.

   If the measurement display area remains unchanged, fast recall is on. To follow along with this explanation, turn fast recall off by pressing **Fast Recall on OFF** so that "OFF" is capitalized.
3. If necessary, select the internal non-volatile RAM disk by pressing **Select Disk** **Non-Vol RAM Disk** .
4. If you have previously saved any files to this disk, they will now be listed on the display.
5. The fast recall feature utilizes only the first seven files listed that contain instrument state, calibration, or measurement data.
6. If you have not previously saved any files to this disk, you may want to save a few instrument states now, to follow along.
7. Press **Prior Menu** **Fast Recall on OFF** so that "ON" is capitalized. Note that the measurement display now reappears on the screen and that there are file names next to the first seven softkeys. (If any of the softkeys are blank, it's because you had less than seven files saved on the disk.)

   The files are placed on the softkeys in the order in which they appear in the disk's directory table.

NOTE

You may want to use the **Rename File** feature to give your files more meaningful names. See "Other File Utilities" in Chapter 4 of your analyzer's *User's Guide* for information on renaming files.

8. To "fast recall" an instrument state, press the softkey next to the file name that contains the instrument state.

9. The fast-recall mode will remain on (even when the analyzer is (PRESET)) until manually turned off.
10. With the fast-recall feature turned on, you will always be only one or two key presses away from recalling an instrument state.
11. With an external keyboard connected to the rear panel DIN connector, keys F1 through F7 are equivalent to pressing softkeys 1 through 7 on the analyzer. See "Using a Keyboard" in Chapter 4 of your analyzer's *User's Guide* for information on connecting and using an external keyboard.

NOTE

For test systems that include a multiport test set:
See the example program titled "MPSVRCL." This IBASIC/RMB program requires an 8712ET/ES or 8714ET/ES analyzer connected to a multiport test set. The program saves three instrument states, each measuring various ports in ALT sweep mode. Fast recall is enabled so that subsequent recalls require a single keypress.

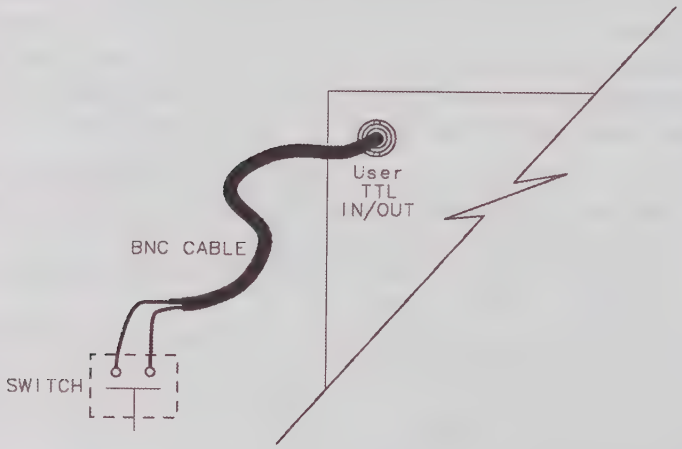
Example programs can be found in the following four locations:
• Web site **http://www.agilent.com**. Use the search function to find Web pages related to 8712 example programs.
• Example Programs Disk, 8712ET/ES and 8714ET/ES (DOS format): part number 08714-10003
• Example Programs Disk, 8712ET/ES and 8714ET/ES (LIF format): part number 08714-10004
• *Example Programs Guide, 8712ET/ES and 8714ET/ES*: part number 08714-90016

Using Fast Recall with a Switch

When fast recall is used in conjunction with a switch connected to the USER TTL IN/OUT rear panel connector, you can cycle through up to seven instrument states in sequence by activating the switch.

1. Connect a switch to the USER TTL IN/OUT rear panel connector as shown in Figure 4-1.

Figure 4-1 Connect a Switch to the USER TTL IN/OUT Connector



po699b

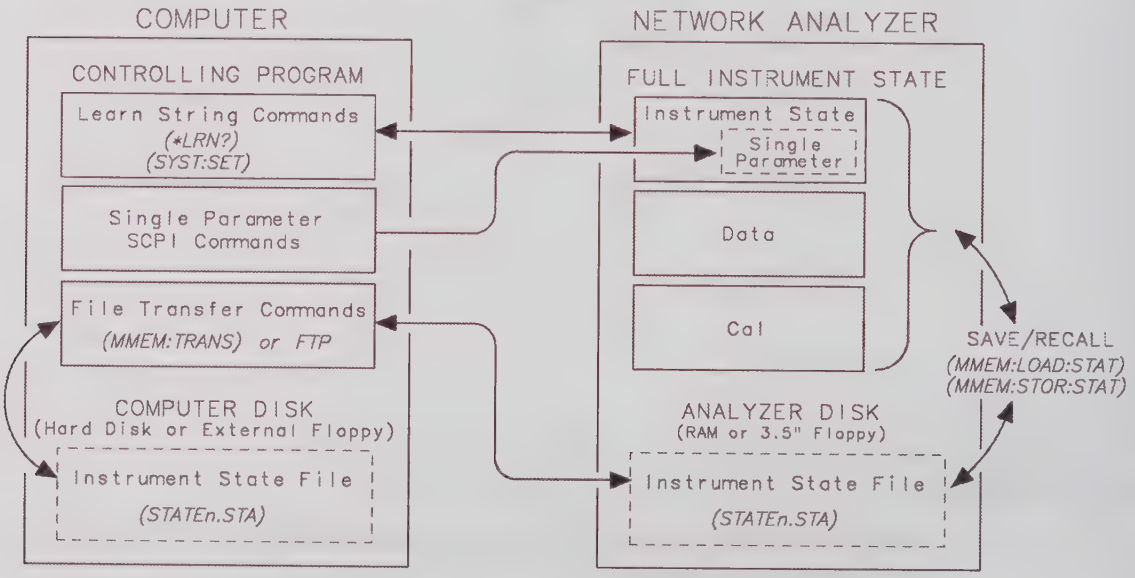
2. Make sure the analyzer is configured to use the USER TTL IN/OUT connector for softkey sequencing: press (SYSTEM OPTIONS) **System Config** **User TTL Config** **Softkey Auto Step** .
3. With **Fast Recall** toggled to ON, press the switch several times while observing the analyzer.
4. Notice that with each press of the switch, the files are highlighted (boxed) in succession, and that after the last available file has been used, the sequence starts again at the top of the softkey menu.

5 Automated Measurement Setup and Control

The production of RF components often involves several steps, with each step requiring a unique set of instrument settings. Likewise, the different test configurations at each step may require associated calibrations. Requiring the operator to manually enter these sets of parameters (or "states") or to calibrate at each step in the manufacturing process is slow, prone to operator error, and costly.

An automated measurement system can be used to achieve fast and consistent transitions between measurement setups. In an automated system, the instrument parameters are set under program control. The control program can be an IBASIC program running inside the analyzer, or a program in another language running on an external computer. The control program sends SCPI and IEEE-488 commands to the analyzer's GPIB or LAN interface. The GPIB commands rapidly change the instrument settings or calibration.

Figure 5-1 **Measurement Control**



The GPIB or LAN interfaces can also be used to trigger sweeps, read measurement values, or signal events within the analyzer. Most operations that can be done from the front panel can also be done over the GPIB or LAN interface. See the *Programmer's Guide* and the *LAN Interface User's Guide Supplement* for details.

This chapter describes methods for changing instrument settings rapidly under program control. It then briefly discusses how to synchronize the modification of instrument settings with the collection of data and how to use service requests (SRQs) to signal instrument states. Finally, it describes how to utilize both measurement channels and a feature to automatically start an IBASIC control program.

For information on techniques that can improve your sweep speed, refer to "Optimizing Measurements" in your analyzer's *User's Guide*.

Setting the Instrument State

A DUT will often undergo several different tests while at a single test station. The analyzer's parameters (such as sweep frequencies, output power, markers, and limits) must be set to the desired values before each test is performed. In an automated test system, the controlling computer modifies the instrument settings for the operator. The analyzer offers several techniques for quickly changing the instrument's measurement parameters:

- Recall of instrument states from disk
- The learn string GPIB command (*LRN)
- SCPI commands that change specific parameters

Recalling Instrument States from Disk

The analyzer has two internal memory (RAM) disks: "Non-Vol RAM Disk" and "Volatile RAM Disk." A "RAM Disk" is a block of memory inside the analyzer which you can access in the same way that you access files on a floppy disk. In addition, the analyzer has a built-in 3.5 inch floppy-disk drive accessible on the front panel.

The "Non-Vol RAM Disk" is non-volatile, meaning that its contents are preserved while the analyzer is turned off. The contents of the "Volatile RAM Disk" are erased when the analyzer is turned off. The volatile RAM disk can be configured to be much larger than the non-volatile RAM disk, allowing it to hold many more instrument states.

The instrument settings and calibrations associated with several tests can be saved to instrument state files on any of the analyzer's three disks described above. The instrument states can later be recalled during a test sequence. The advantage of using RAM disks rather than the 3.5 inch floppy disk is that recalling a state from RAM disks takes several seconds less than recalling a state from the floppy disk.

For example, suppose the instrument settings are entered for the third test of a sequence of tests. The instrument settings can be saved in a file with the name "TEST3.STA" on the non-volatile RAM disk. When the third test must be performed, the control program can recall the instrument state from the non-volatile RAM disk with the following SCPI command:

```
MMEM:LOAD:STATe 1,'MEM:TEST3.STA'
```

Recalling a state from RAM disk typically takes about 4 seconds, but the time is dependent on settings such as number of points.

One strategy for managing a large set of recall states is to initially store them on a floppy disk. The files can then be copied from the floppy disk to the volatile RAM disk at the start of each day. Since recall time from the RAM disk is faster than recall from the floppy disk drive, throughput can be increased. For example, to copy the file 'TEST3.STA' from the 3.5 inch disk to the non-volatile RAM disk, use the following SCPI command:

```
MMEM:COPY 'INT:TEST3.STA', 'MEM:'
```

For more details, see the *Programmer's Guide*.

When controlling the analyzer with a computer, you may want to copy an instrument state file from your computer to one of the analyzer's built-in RAM disks or the 3.5 inch disk. Later, the instrument can be instructed to recall the state from its internal disk. The file can be sent from the computer to the analyzer's disk using the SCPI `MMEMory:TRANsfer` commands. For more details, see the *Programmer's Guide*.

For manually controlled test systems, the instrument state files for each measurement can be stored onto disk by the test-system designer. During the test sequence, the operator can press the SAVE RECALL key and recall the instrument state for a particular measurement.

Using Learn Strings to Save and Recall Instrument States

The IEEE 488 *LRN ("learn") command can also be used to set or query a complete set of instrument parameters. This can be used as a programming convenience, eliminating the need for using disk files when saving and recalling instrument states.

To obtain the learn string containing the instrument state, use the command *LRN? as follows:

```
10  DIM State1$[4000]
20  OUTPUT @Rfna;"*LRN?"
30  ENTER @Rfna USING "-K";State1$
    ...
160  ! Put the learn string back
170  OUTPUT @Rfna;State1$
```

Since *LRN?, by IEEE definition, only contains the actual instrument state, exclusive of data traces and calibration arrays, the network analyzer provides the command

```
SYST:SET:LRNLong?
```

This command saves the data traces and calibration arrays if they are enabled using (SAVE RECALL) **Define Save** or by using the MMEM:STOR:STATE SCPI commands. Using the *LRN command to set the instrument settings takes about the same amount of time as recalling a file from disk using MMEM:LOAD:STATe.

For more details on learn strings, refer to the *Example Programs Guide.*

SCPI Commands that Modify a Single Parameter

TIP

If several measurement setups are similar — differing only by a few instrument parameters — the fastest way to switch between the states is for the control program to send the SCPI commands that modify only those parameters.

For example, if center frequency and source power are the only parameters that change in consecutive measurement setups, send the SCPI commands `SENS1:FREQ:CENT` and `SOUR1:POW` to change these parameters, leaving all other instrument settings unchanged. This will be faster than either recalling an instrument state or sending a learn string.

Fast Iterative Control

You may want to quickly and iteratively change the analyzer's parameters and monitor their effect on your device under test. This section describes how to quickly change a parameter, take a sweep, and query a marker.

For example, when measuring an amplifier, you may wish to quickly choose the optimal input power to the amplifier which will result in a maximum output of +10 dBm. To do so, you can first set the analyzer's source power level, then measure the amplifier's output, and then change the analyzer's source power in the direction that will cause the measured signal to approach the desired value. This process can be repeated until the measured amplifier output is within some specified range of the target value.

When data at only a single frequency is needed, you can achieve the fastest possible sweep by selecting a CW frequency and setting the number of points to the minimum value of 3. A SCPI marker command can be used to read the trace value.

Using this approach, you can typically achieve 3 to 5 sweeps per second. Following is a listing of an example program named "FAST_CW" that can be found on your *Example Programs Disk.*

```
100   DIM Freq_str$[20]
110   DIM Msg$[100]
120   !
130   IF POS(SYSTEM$("SYSTEM ID"),"HP 871") THEN
140     ASSIGN @Hp8711 TO 800
150   ELSE
160     ASSIGN @Hp8711 TO 716
170     ABORT 7
180     CLEAR 716
190   END IF
200   !
210   ! PRESET, to ensure known state.
220   OUTPUT @Hp8711;"SYST:PRES;*WAI"
230   CLS
240   !
250   ! Set up the analyzer to measure 3 data points.
260   OUTPUT @Hp8711;"SENS1:SWE:POIN 3;*WAI"
270   !
280   ! Select CW display and sweep
290   OUTPUT @Hp8711;"DISP:ANN:FREQ1:MODE CW"
300   OUTPUT @Hp8711;"SENS1:FREQ:SPAN 0 Hz;*WAI"
310   !
320   ! Take a single sweep, leaving the analyzer
330   ! in trigger hold mode.
```

```
340   OUTPUT @Hp8711;"ABOR;:INIT1:CONT OFF;*WAI"
350   !
360   ! Turn on Marker 1
370   OUTPUT @Hp8711;"CALC:MARK1 ON"
380   !
390   Count=0
400   T0=TIMEDATE
410   ! Step from 175 MHz 463 MHz by 6 MHz
420   FOR Freq=175 TO 463 STEP 6
430     ! Take a sweep
440     Freq_str$=VAL$(Freq)&" MHz"
450     OUTPUT @Hp8711;"SENS1:FREQ:CENT ";Freq_str$
460     OUTPUT @Hp8711;"INIT1;*WAI"
470     !
480     ! Set marker to frequency
490     OUTPUT @Hp8711;"CALC:MARK:X ";Freq_str$
500     !
510     ! Query the marker value
520     OUTPUT @Hp8711;"CALC:MARK:Y?"
530     ENTER @Hp8711;Response
540     !
550     ! Display the first three numbers in the array.
560     Msg$="'"&Freq_str$&": "&VAL$(Response)&"'"
570     OUTPUT @Hp8711;"DISP:ANN:MESS ";Msg$
580      PRINT Msg$
590      Count=Count+1
600   NEXT Freq
610   T1=TIMEDATE
620   PRINT "Sweeps per second: ";Count/(T1-T0)
630   DISP "Sweeps per second: ";Count/(T1-T0)
640   END
```

Responsive Communication using SRQs

Service requests (SRQs) are a method by which you can instruct the analyzer to tell your computer program when a condition changes or when an event of interest occurs. This communication is done via GPIB signals.

Analyzer SRQ events include:

- Limit test fails
- A front panel key or external keyboard key is pressed
- Hardcopy in progress or complete
- Sweep in progress or complete
- Power has been cycled

The analyzer can be set to cause an SRQ on any combination of the above events. Using SRQs allows your program to be interrupt driven, reducing the latency and inefficiency of polling. For more details, refer to "Using Status Registers" in the *Programmer's Guide.*

NOTE

For test systems that include a multiport test set:
See the example program titled "MPCALSRQ." This IBASIC/RMB program requires an 8712ET/ES or 8714ET/ES analyzer connected to a multiport test set. The program configures the analyzer's status reporting so that an SRQ is issued whenever a SelfCal is initiated.

Example programs can be found in the following four locations:
• Web site **http://www.agilent.com**. Use the search function to find Web pages related to 8712 example programs.
• Example Programs Disk, 8712ET/ES and 8714ET/ES (DOS format): part number 08714-10003
• Example Programs Disk, 8712ET/ES and 8714ET/ES (LIF format):part number 08714-10004
• *Example Programs Guide, 8712ET/ES and 8714ET/ES*: part number 08714-90016

Using Both of the Analyzer's Measurement Channels

The analyzer is capable of making different measurements on each of its two measurement channels. For example, you can set measurement channel 1 to measure transmission over one set of sweep frequencies, while measurement channel 2 is set to measure reflection over another set of sweep frequencies. Thus, two measurements can be made by the operator at the same time. Also, the controller can switch between measurement channel 1 and measurement channel 2, while turning the inactive channel off, to quickly change the test setup between two test states.

Using Single Sweep to Measure and View All Four S-Parameters

You can save sweep time when measuring S-parameters on an 8712ES or 8714ES by using the single sweep feature. As long as a full two-port calibration has been performed, you can measure all four S-parameters with a single sweep cycle, and then individually view each S-parameter by then pressing the desired S-parameter softkey under the (MEAS 1) or (MEAS 2) menu.

From the front panel, set up your measurement parameters, perform (or recall) a valid full two-port calibration, and then connect the DUT. Press (MENU) **Trigger Single** . The analyzer takes a forward and reverse sweep to capture all four S-parameters, and then goes into "hold sweep" mode. Press (MEAS 1) or (MEAS 2), and then any of the S-parameter measurement keys to view the desired parameter.

The SCPI command to perform a single sweep and then hold is

```
ABOR;:INIT[1|2]:CONT OFF;:INIT[1|2]; *WAI
```

For an automated example, see the example program listed below. This program is also available titled "FOURPARM" on your *Example Programs Disk*.

```
1000 !Filename: FOURPARM
1010 !
1020 ! Demonstrates that all four S-parameters can be computed by
1030 ! the instrument after a single sweep.
1040 !
1050 ! Outline:
1060 ! 1. Select Default 2-Port Cal
1070 ! 2. Select S11 on MEAS 1. S21 on MEAS 2.
1080 ! These are Port 1 Reflection and forward transmission.
1090 ! Take a sweep.
1100 ! 3. Do Data->Mem on MEAS 1 and on MEAS 2.
1110 ! 4. Without taking another sweep, select
1120 ! S22 on MEAS 1 and S12 on MEAS 2.
1130 ! These are Port 2 Reflection and reverse transmission.
1140 !
1150 ! The display will show:
1160 !
1170 ! MEAS 1 : Port 1 and Port 2 REFLECTION (S11 and S22)
1180 ! MEAS 2 : forward and reverse TRANSMISSION (S21 and S12)
1190 !
1200 ! Note that a single MEAS channel could be used to gather all
four
1210 ! S-parameters after a single sweep. For display convenience
1220 ! however, this program uses both MEAS channels.
```

```
1230 !
1240 DIM A$[10],Data1(1:201)
1250 INTEGER Digits,Bytes
1260 COM /Sys_state/ @Hp87xx,Scode
1270 !
1280 CALL Iden_port ! Identify I/O Port
1290 !
1300 OUTPUT @Hp87xx;"SYST:PRES" ! System Preset
1310 OUTPUT @Hp87xx;"*OPC?"
1320 ENTER @Hp87xx;Opc_flag ! Operation complete
1330 !
1340 ! Select Default Two Port Calibration on MEAS 1 and 2
1350 !
1360 DISP "Enabling 2-Port Calibration..."
1370 OUTPUT @Hp87xx;"SENS2:STAT ON" ! Turn on MEAS 2
1380 OUTPUT @Hp87xx;"SENS1:FUNC 'XFR:S 1,1'" ! MEAS 1 = S11
1390 OUTPUT @Hp87xx;"SENS2:FUNC 'XFR:S 2,1'" ! MEAS 2 = S21
1400 OUTPUT @Hp87xx;"SENS1:CORR:CLASS:SEL DEFAULT2" ! Default
2-Port
1410 OUTPUT @Hp87xx;"SENS2:CORR:CLASS:SEL DEFAULT2" ! Default
2-Port
1420 OUTPUT @Hp87xx;"DISP:FORM ULOW" ! Split Display
1430 OUTPUT @Hp87xx;"*OPC?"
1440 ENTER @Hp87xx;Opc_flag ! Operation complete
1450 DISP ""
1460 !
1470 ! Take a single sweep, leaving the analyzer
1480 ! in trigger hold mode.
1490 !
1500 OUTPUT @Hp87xx;"ABOR;:INIT1:CONT OFF;:INIT1;*WAI"
1510 !
1520 ! Copy MEAS 1 Data (S11) into MEAS 1 Memory trace
1530 ! Copy MEAS 2 Data (S21) into MEAS 2 Memory trace
1540 !
1550 DISP "Copy S11 to Mem1"
1560 OUTPUT @Hp87xx;"TRAC CH1SMEM,CH1SDATA" ! Ch1 Data->Mem
1570 DISP "Copy S21 to Mem2"
1580 OUTPUT @Hp87xx;"TRAC CH2SMEM,CH2SDATA" ! Ch2 Data-> Mem
1590 OUTPUT @Hp87xx;"*OPC?"
1600 ENTER @Hp87xx;Opc_flag
1610 !
1620 ! Now select S22 on MEAS 1 and S12 on MEAS 2.
1630 !
1640 DISP "Meas S22 on ch1"
1650 OUTPUT @Hp87xx;"SENS1:FUNC 'XFR:S 2,2'" ! MEAS 1 = S22
1660 OUTPUT @Hp87xx;"*OPC?"
1670 ENTER @Hp87xx;Opc_flag ! Opeation complete.
1680 !
1690 DISP "Meas S12 on ch2"
1700 OUTPUT @Hp87xx;"SENS2:FUNC 'XFR:S 1,2'" ! MEAS 2 = S12
1710 OUTPUT @Hp87xx;"*OPC?"
1720 ENTER @Hp87xx;Opc_flag ! Opeation complete.
1730 !
1740 ! Display Data and Memory on both MEAS channels. Autoscale
result.
1750 !
1760 OUTPUT @Hp87xx;"CALC1:MATH (IMPL);:DISP:WIND1:TRAC1 ON;TRAC2
ON"
1770 OUTPUT @Hp87xx;"DISP:WIND1:TRAC:Y:AUTO ONCE"
1780 OUTPUT @Hp87xx;"CALC2:MATH (IMPL);:DISP:WIND2:TRAC1 ON;TRAC2
```

```
ON"
1790 OUTPUT @Hp87xx;"DISP:WIND2:TRAC:Y:AUTO ONCE"
1800 !
1810 DISP "Ch1: S11 (M:S22) Ch2: S12 (M:S21)" ! Done
1820 BEEP
1830 !
1840 END
1850 !
1860 !
1870
!*****************************************************************
1880 ! Iden_port: Identify io port to use
1890 ! Description: This routines sets up the I/O port address for
1900 ! the SCPI interface. For "HP 87xx" instruments,
1910 ! the address assigned to @Hp87xx = 800 otherwise,
1920 ! 716.
1930
!*****************************************************************
1940 SUB Iden_port
1950 COM /Sys_state/ @Hp87xx,Scode
1960 !
1970 IF POS(SYSTEM$("SYSTEM ID"),"HP 87")<>0 THEN
1980 ASSIGN @Hp87xx TO 800
1990 Scode=8
2000 ELSE
2010 ASSIGN @Hp87xx TO 716
2020 Scode=7
2030 END IF
2040 !
2050 SUBEND !Iden_port
2060 !
```

AUTOST files

When IBASIC is used, the measurement control program can be saved as an AUTOST file on the analyzer's non-volatile RAM disk. When the analyzer's power is turned on, it will first check for this file on the non-volatile RAM disk and then on the 3.5 inch disk, and if found, load it and run it. This feature simplifies the task of turning on an automated test station at the beginning of a working day or test session. To manually load and run an autostart file, press (BEGIN) **Autost** .

6 Controlling Peripherals

You can access the analyzer's rear panel interface ports from your measurement control program. Using this capability, you can communicate with peripherals such as material handlers, custom DUT interface circuits, external switch boxes, and printers.

This chapter will focus on use of the Centronics parallel port and the RS-232 serial port. Communication with the DIN KEYBOARD interface, the USER TTL, and the LIMIT TEST TTL connectors is described in detail in Chapter 3, "Operator Interaction."

NOTE

Refer to your analyzer's *User's Guide* for pinout illustrations and descriptions of the parallel and serial ports.

Using the Parallel Port

The analyzer's parallel (Centronics) port can be used as an 8-bit TTL output port and as a 5-bit TTL input port. The eight TTL outputs are for output only, and cannot be read or used as bi-directional I/O lines. The parallel port does *not* support the IEEE-1284-defined Extended Capabilities Port (ECP) mode or Enhanced Parallel Port (EPP) mode.

The output signals are driven by standard TTL drivers. They should be buffered for heavy-duty applications to avoid damaging the analyzer. The inputs are standard TTL inputs, designed to accept signals in the 0 V to 5 V range.

The analyzer provides two ways to access the parallel port. You can use the SCPI commands

```
OUTPUT @Rfna;"DIAG:PORT:WRITE <port>,<register>,<data>
```

and

```
OUTPUT @Rfna;"DIAG:PORT:READ? <port>,<register>
ENTER @Rfna;Data
```

or you can use IBASIC and its `READIO` and `WRITEIO` commands. See the following tables for more information.

Table 6-1 Writeable Ports

Port Number	Register	Description
15	0	Outputs 8-bit data to the Cent_D0 thru D7 lines of the Centronics port. Cent_D0 is the least significant bit, Cent_D7 is the most significant bit. Checks Centronics status lines for: Out of Paper Printer Not on Line BUSY ACKNOWLEDGE
15	1	Sets/clears the user bit according to the least significant bit of A. A least significant bit equal to 1 sets the user bit high. A least significant bit of 0 clears the user bit.
15	2	Sets/clears the limit pass/fail bit according to the least significant bit of A. A least significant bit equal to 1 sets the pass/fail bit high. A least significant bit of 0 clears the pass/fail bit.
15	3	Outputs 8-bit data to the Cent_D0 thru D7 lines of the Centronics port. Cent_D0 is the least significant bit, Cent_D7 is the most significant bit. Sets the Printer_select signal high (de-select). Does not check Centronics status lines.
9	0	Outputs a byte to the serial port. The byte is output serially according to the configuration for the serial port.

Table 6-2 Readable Ports

Port Number	Register	Description
9	0	Reads the serial port.
15	0	Reads the 8-bit data port Cent_D0 thru D7.
15	1	Reads the user bit.
15	2	Reads the limit test pass/fail bit.
15	10	Reads the 8-bit status port. D0 — Cent_acknowledge D1 — Cent_busy D2 — Cent_out_of_paper D3 — Cent_on_line D4 — Cent_printer_err

NOTE When using the `WRITEIO(15,0)` or `WRITEIO(15,3)` command, the Printer_Select Line is set high. However, when the instrument is doing hardcopy, the Printer_Select Line is set low. The Printer_Select line may or may not be used by individual printers. Check with your printer manual.

Writing to the Parallel Port

To write the value 52 decimal (34 hex, 0011 0100 binary) to the parallel port's output pins, use one of the following commands:

```
OUTPUT @Rfna;"DIAG:PORT:WRITE 15,3,52"
WRITEIO 15,3;52
```

When the write command is executed, the parallel port's data lines (pins 2-9) will be set to the specified value, and then a pulse of at least 1 µs duration will occur on the strobe line (pin 1). A data setup and data hold time of at least 1 µs are guaranteed. See Figure 6-1.

Figure 6-1 **Writing to the Parallel Port**

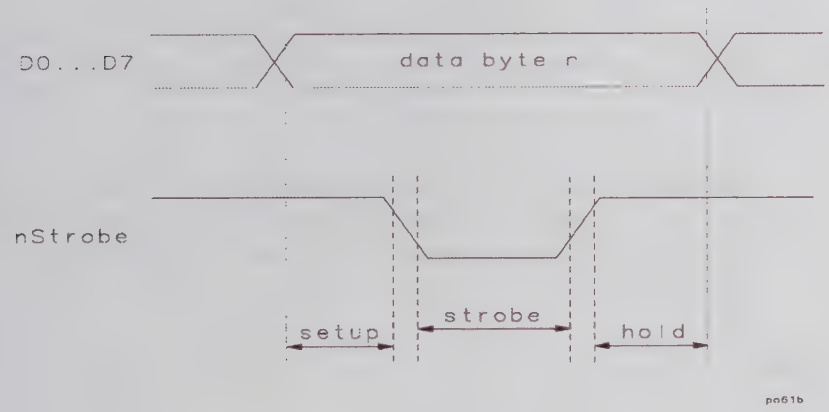


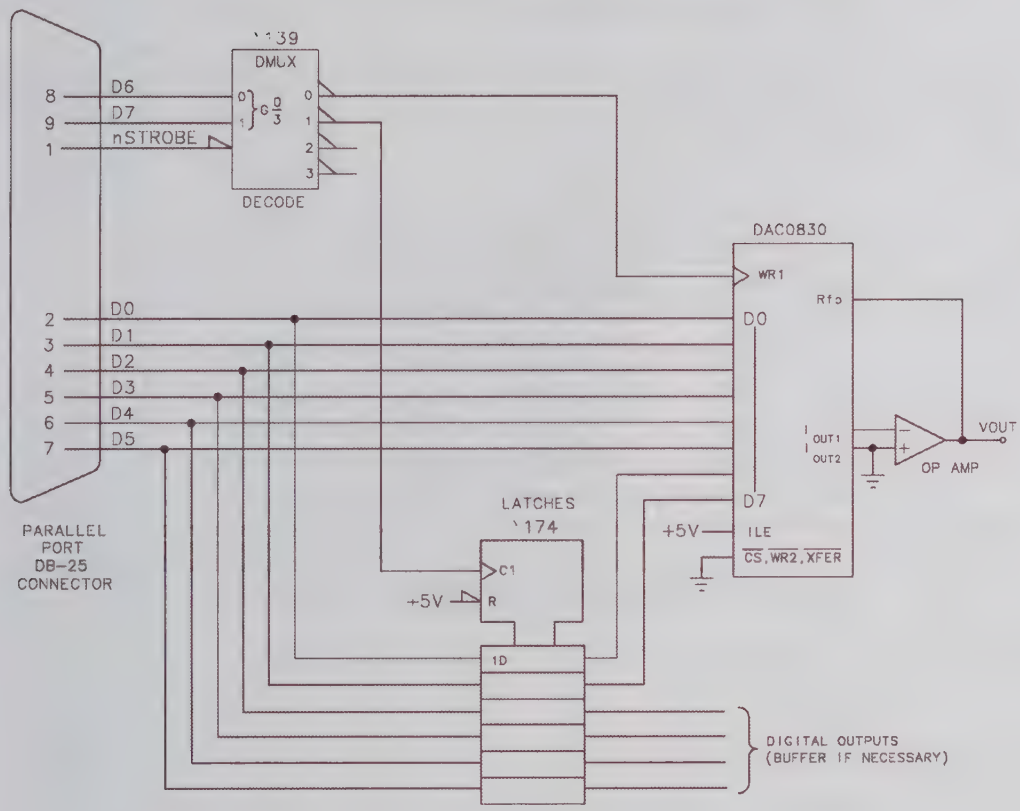
Table 6-3 shows the pin numbers, data bus bit numbers, and signal names:

Table 6-3 Parallel Port Pins

Pin	Bit	Name
1	N/A	Strobe
2	D0	Data 1
3	D1	Data 2
4	D2	Data 3
5	D3	Data 4
6	D4	Data 5
7	D5	Data 6
8	D6	Data 7
9	D7	Data 8

The data will typically remain valid until the next write to the parallel port, but you should always latch the data using the strobe. Figure 6-2 shows a simple circuit which can be used to write to an 8-bit DAC and a digital latch.

Figure 6-2 **Digital Latch Circuit**



NOTE: ESD PROTECTION, POWER SUPPLIES AND DECOUPLING ARE NOT SHOWN.

po673b

Reading from the Parallel Port

The parallel port has five TTL input signals, normally used for determining the printer's status, which can be read. The signals and the corresponding data bits and pins are shown in the following table:

Table 6-4 **Parallel Port Inputs**

Pin	Bit	Name
10	D0	Acknowledge
11	D1	Busy
12	D2	Out of Paper
13	D3	On Line
15	D4	Printer Error

Your custom interface circuit can drive these signals, and they can be read using the following commands.

SCPI commands:

```
OUTPUT @Rfna;"DIAG:PORT:READ? 15,10"
ENTER @Rfna;Parallel_in
```

IBASIC command:

```
Parallel_in = READIO(15,10)
```

Hardcopy Considerations

The analyzer's (HARDCOPY) feature can send output to printers connected to the parallel port. If you have a custom interface circuit attached to the parallel port, you don't want the hardcopy output to interfere with it.

To address this issue, the analyzer uses the parallel port's printer-select signal (pin 17) to differentiate between hardcopy dumps and user-issued `WRITEIO` and `DIAG:PORT:WRITe` commands. During a hardcopy to the parallel port, the printer-select signal is driven low. During `WRITEIO` and `DIAG:PORT:WRITe` commands, it is driven high.

Using the printer-select signal, you can connect both an interface circuit and a printer to the parallel port. The interface circuit should only respond to the data strobe (pin 1) when the printer-select signal is high. The printer should only respond to the data strobe when the printer-select signal is low. Most printers require the printer-select line to be low, or else they will not print. Some printers ignore this line. If you are using a printer which ignores this line you have two choices:

1. Consult the printer's manual to see if there is a DIP-switch setting that controls how the printer responds to the printer-select signal, or if not,
2. Design your interface circuit so that it gates and inhibits the data strobe signal (pin 1) going to the printer when printer-select is high.

Similar to the output signals, you can use the printer-select signal to multiplex the input signals, selecting either the signals from your interface circuit or those from the printer.

Using the Serial Port

Like the parallel port, the RS-232 serial port can also be accessed using SCPI and IBASIC commands.

To write a byte with a value of 52 decimal (34 hex, 0011 0100 binary) to the serial port, use one of the following commands:

```
OUTPUT @Rfna;"DIAG:PORT:WRITE 9,0,52"
WRITEIO 9,0;52
```

To read a byte from the serial port, use the following commands:

```
OUTPUT @Rfna;"DIAG:PORT:READ? 9,0"
ENTER @Rfna;Serial_in
```

If you are using IBASIC, you can simply use the `READIO` statement:

```
Serial_in = READIO(9,0)
```

For general purpose I/O, the parallel port is much easier to interface to than the serial port. To interface to the serial port, a Universal Asynchronous Receiver Transmitter (UART) is typically used to decode the RS-232 signals. Most UARTs are designed to be used with microprocessors.

The advantage of the serial port is that it can operate over long distances: up to 30 meters using the RS-232-C standard. Its disadvantage is its slow speed which is limited to 19200 bits/second.

Before using the serial port, you must select the baud rate and handshake style using the SCPI commands:

```
SYSTem:COMMunicate:SERial:TRANsmit:BAUD
SYSTem:COMMunicate:SERial:TRANsmit:HANDshake {XON|DTR}
```

One type of application for the serial port is data logging to a remote computer. After each device is measured, an IBASIC program could use the `WRITEIO` command shown above to send a brief summary of the measurement result, such as a filter's 3 dB bandwidth and its serial number, to a remote computer. A program on the remote computer would monitor the serial port and read the incoming data and archive it to a hard disk or network.

WARNING

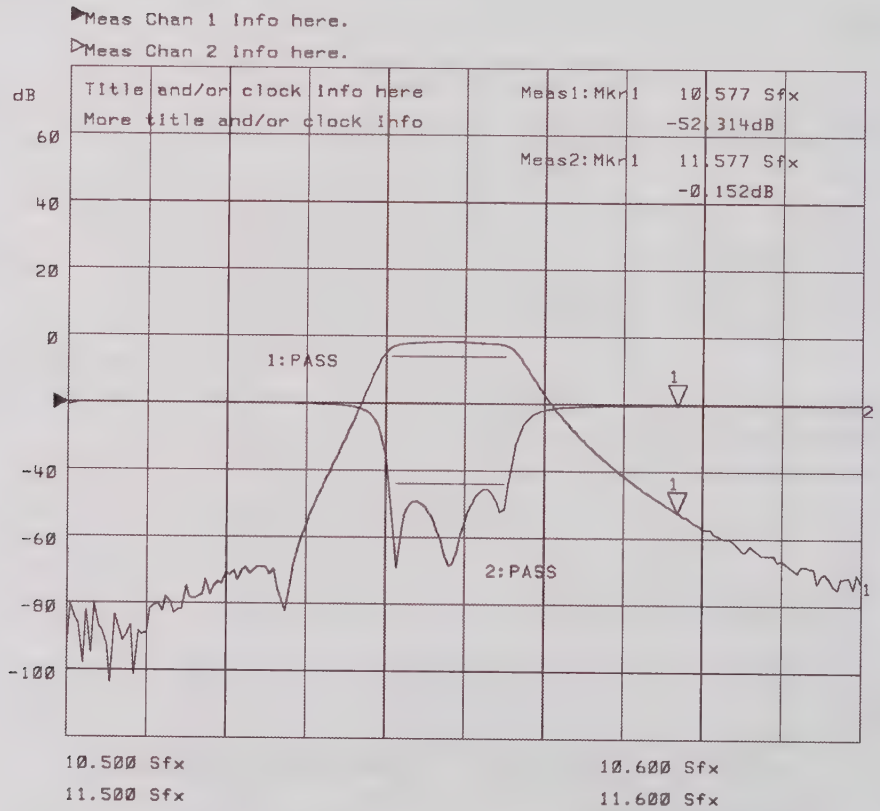
The handshaking on the serial port is very limited. It is not possible to read the status of the serial port buffers (cannot detect when an input character has been received or when the output buffer is empty).

7 Displaying Measurement Results

It is often helpful to eliminate unnecessary and potentially distracting information and annotation from the display, and only show the information necessary to perform a particular task.

The analyzer provides several features to let you customize the information shown on the display as shown in Figure 7-1. Many of the features discussed in this chapter can also be implemented using the analyzer's user interface. See "Customizing the Display" in Chapter 4 of your analyzer's *User's Guide* for more information.

Figure 7-1 Customized Annotation



Customizing features such as limit testing, annotating the X-axis, and using the title feature are described in this chapter.

Graticule On/Off

The graticule is the set of grid lines that designate increments of value on the X-axis and Y-axis of the measurement.

If you are comparing the trace against limit lines or marker values, turning off the graticule makes it easier to view the measurement trace, limit lines, and markers. To turn the graticule off, press DISPLAY **More Display Graticule ON off** or use the following SCPI command:

```
DISPlay:WINDow[1|2]:TRACe:GRATicule:GRID OFF
```

where the window number is 1 if in full-screen display, and 1 or 2 for the upper and lower split-screen displays.

Limit Testing

The measurement trace can be automatically compared to limits which you define. The limits, entered as lines and points, can be displayed on the screen or can be hidden. Whether or not the limits are displayed, the analyzer will display "PASS" if the measurement satisfies the limits and will display the FAIL ⚠ symbol if the measurement fails to meet the prescribed limits. Using limits gives the operator visual guides when tuning devices, provides standard criteria for meeting device specifications, and shows an instant indication of the comparison of data versus specifications.

To turn limit lines on or off

To turn limit lines on or off, press (DISPLAY) **Limit Menu** **Limit Options** **Limit Line ON off**, or use the following SCPI commands:

```
CALC[1|2]:LIMit:DISPlay ON
CALC[1|2]:LIMit:DISPlay OFF
```

where [1 | 2] indicates the measurement channel number; use either 1 or 2.

To turn the fail icon on or off

To turn the fail icon on or off, press (DISPLAY) **Limit Menu** **Limit Options** **Limit Icon ON off**, or use the following SCPI commands:

```
DISP:ANN:LIM:ICON[1|2]:FLAG ON
DISP:ANN:LIM:ICON[1|2]:FLAG OFF
```

To turn the pass/fail text on or off

To turn the pass/fail text on or off, press (DISPLAY) **Limit Menu** **Limit Options** **Limit Text ON off**, or use the following SCPI commands:

```
DISP:ANN:LIM:ICON[1|2]:TEXT ON
DISP:ANN:LIM:ICON[1|2]:TEXT OFF
```

To move the position of the pass/fail indicator

To move the position of the pass/fail indicator, press DISPLAY **Limit Menu** **Limit Options**, and use the **Limit Icon X Position** and **Limit Icon Y Position** keys to reposition the indicator. Or use the following SCPI commands:

```
DISP:ANN:LIM:ICON[1|2]:POS:X <num>
DISP:ANN:LIM:ICON[1|2]:POS:Y <num>
```

For more information on limit lines, see "Using Limit Testing" in Chapter 4 of your analyzer's *User's Guide*.

See Figure 7-1 for an example of a measurement using limit lines with a "PASS" test result.

Customized X-axis Annotation

X-axis annotation consists of one or two lines of information that appear below the graticule. You can replace this annotation with your own text (see Figure 7-1) or eliminate the X-axis annotation completely.

By default, the X-axis annotation displays the stimulus frequencies (default resolution is kHz), or powers if in power sweep. However, it can be customized using SCPI commands to show your own start and stop X-axis values and units. For example, when measuring mixers which introduce a frequency offset, you can annotate the frequencies at the output of the mixer.

To turn on user-defined X-axis annotation

To turn on user-defined X-axis annotation, use the command:

```
DISP:ANN:FREQ[1|2]:USER[:STATe] {OFF|0|ON|1}
```

For example:

```
DISP:ANN:FREQ1:USER ON
```

To specify your start and stop values

To specify your start and stop values, use:

```
DISP:ANN:FREQ[1|2]:USER:STARt <num>
DISP:ANN:FREQ[1|2]:USER:STOP <num>
```

The value <num> must be between –10,000 and 10, 000. For example:

```
DISP:ANN:FREQ1:USER:STAR -100
DISP:ANN:FREQ1:USER:STOP +100
```

To specify a custom suffix

To specify a custom suffix, use:

```
DISP:ANN:FREQ[1|2]:USER:SUFF[:DATA]
```

For example:

```
DISP:ANN:FREQ1:USER:SUFF 'uV'
```

The suffix can be up to 3 characters long.

To turn the X-axis annotation on or off

To turn the X-axis (frequency) annotation on or off, press DISPLAY **More Display Annotation Options Freq Annot ON off** or use the following SCPI commands:

```
DISP:ANN:FREQ[1|2] ON

DISP:ANN:FREQ[1|2] OFF
```

NOTE

When using custom X-axis annotation, the SCPI command `CALC:MARK:X` and query `CALC:MARK:X?` will return the analyzer's stimulus value, *not* your custom annotation values. If this is a problem, you can use the SCPI command `CALC:MARK:POIN` to specify the X-axis point number at which you wish to position the marker.

For example:

```
OUTPUT @HP8711;"CALC:MARK1:POIN 134"
```

will put the marker at point number 134.

Custom X-axis annotation has no effect on Y-axis marker values (`CALC:MARK:Y?`).

Customized Measurement-Channel Annotation

The analyzer displays measurement-channel annotation above the graticule. This annotation shows the measurement type, format, scale/division, and reference level.

You may replace this annotation with your own text (see Figure 7-1) or eliminate the measurement-channel annotation completely.

To turn on user-defined measurement-channel annotation

To turn on user-defined measurement-channel annotation, use the command:

```
DISP:ANN:CHAN[1|2]:USER[:STATe] ON
```

where `[1|2]` indicates the measurement-channel number: either 1 or 2.

To specify the string to be displayed

To specify the string to be displayed use the command:

```
DISP:ANN:CHAN[1|2]:USER:LAB[:DATA]
```

For example:

```
DISP:ANN:CHAN1:USER:LABel '1: SuperNotch filter, test #3'
```

To restore default measurement-channel annotation

To restore the default measurement-channel annotation, use:

```
DISP:ANN:CHAN[1|2]:USER OFF
```

To turn measurement-channel annotation on or off

To turn the measurement-channel annotation on or off, press DISPLAY **More Display Annotation Options Meas Annot ON off**, or use the following SCPI commands:

```
DISP:ANN:CHAN[1|2] ON
DISP:ANN:CHAN[1|2] OFF
```

Markers

Well-designed use of the analyzer's marker features in an automated system can enhance the data gathering process, as well as provide important visual feedback to the operator.

The active marker's value is displayed in the upper right area of the graticule. If marker bandwidth (or notch) search is selected, the bandwidth (or notch) information is displayed instead. This marker information can be used to view exact measured data at critical frequency points. Note the customized measurement channel notation in Figure 7-1. In addition to the active marker's readout, four of the marker's values are displayed in the softkey area during front panel use. This makes it easy to quickly read the measured data at several marker positions.

The triangular marker symbols can also be used to graphically indicate critical frequency points of the measurement. For example, a marker can be set at the desired center frequency for a notch filter, and the operator can tune the filter until the notch is at the same frequency as the marker.

Marker search types include:

- max search
- min search
- target search
- bandwidth
- notch
- multi peak
- multi notch

When marker tracking is turned on, these searches will be automatically performed at the end of each sweep. This can be useful in tuning applications.

Other marker functions that can be useful are the marker math functions:

- statistics
- flatness
- RF filter stats

These functions perform certain mathematical calculations on the amplitude, phase, delay or SWR data of user-defined trace segments. See "To Use Marker Math Functions" in Chapter 4 of your analyzer's *User's Guide* for more information on these features.

Title and Clock

The analyzer has two 30-character title lines. One of these lines can be replaced with a real-time clock readout. Timestamps on measurement data can be important when monitoring throughput, quality, and processes.

For example, the title line can be set to show the serial number and type of the DUT. Doing so provides a simple and safe link between the device under test and the measurement data.

The title and clock lines are, by default, included on hardcopy printouts. These can be configured using the (DISPLAY) **More Display Title and Clock Title Line 1** menu. For more details and a simple example, refer to "Using the Analyzer's Title Feature" on page 3-15.

8 Saving Measurement Results

After measuring a device, you will probably want to save the measurement results so that you can perform statistical analyses on them. Statistical quality control (SQC) can be a powerful tool to indicate process drift or variation.

You may also want to produce a print or plot of the DUT's response, and ship this to your customer along with the DUT.

Querying Measurement Data

To save the complete measurement trace, use the SCPI command:

```
CALC1:DATA?
```

or

```
TRACE1:DATA? CH1FDATA
```

Refer to the chapter titled "Trace Data Transfers" in the *Programmer's Guide* for more details.

From IBASIC, you can also use the subprogram "Read_fdata()", which is faster. Refer to the chapter titled "Using Subprograms" in the *HP Instrument BASIC User's Handbook Supplement.*

Saving the Measurement to Disk — Save ASCII

The analyzer has a **Save ASCII** feature which saves the measurement trace in a format compatible with many popular spreadsheet programs such as Lotus® 1-2-3® and Microsoft Excel®. The measurement can also be saved in Touchstone (S2P) format for importing data into CAE programs. The measurement is saved to a file on the analyzer's disk.

The following program segment shows how to save the measurement in Lotus 1-2-3 format to a file on the analyzer's non-volatile RAM disk, and then how to transfer that file into your program and store it as a file on your computer disk.

```
10 DIM A$[32000]
20 Dest$="SAV_DUT1.PRN"
30 OUTPUT 716;"MMEM:STOR:TRAC:FORMAT LOTUS123"
40 OUTPUT 716;"MMEM:STOR:TRAC CH1FDATA, 'MEM:DUT1.PRN'"
50 OUTPUT 716;"MMEM:TRAN? 'MEM:DUT1.PRN'"
60 ENTER 716 USING "W,-K";Word1,A$
70 CREATE Dest$,32000
80 ASSIGN @File TO Dest$
90 OUTPUT @File;A$
100 ASSIGN @File TO *
```

With the **Save ASCII** feature, you can read the measurements into your CAE or spreadsheet program, and perform statistical analysis on the data, such as mean and standard deviation on groups of DUTs.

For information on transferring disk files between the analyzer and your computer, refer to the *Example Programs Guide.*

Saving the Measurement to Disk — Save Data

Similar to **Save ASCII** , the analyzer can also save the measurement data onto disk as an instrument state file. Use **Define Save** , and turn Data ON and turn Inst State and Cal OFF.

NOTE

A file saved in this manner is smaller than a file saved using **Save ASCII** . **Save Data** uses 6 bytes per point, as opposed to about 20 bytes per point for **Save ASCII** . However, the file type is binary, and contains a header, making it difficult to read.

Querying Marker Searches

The analyzer can measure a filter and compute its center frequency, bandwidth, Q, and loss. You can query this information using the SCPI command

```
CALC:MARK:FUNC:RES?
```

For example:

```
10 OUTPUT @Rfna;"CALC:MARK:BWID -3"      ! -3 dB bandwidth
20 OUTPUT @Rfna;"CALC:MARK:FUNC:RES?"    ! Get result of bandwi
dth search
30 ENTER @Rfna;Bwidth,Center_freq,Q,Loss
```

For more details, refer to the *Example Programs Guide*.

Saving Measurement Results to Disk

The analyzer provides two internal RAM disks and one internal 3.5 inch floppy disk to which measurement results can be saved.

For the fastest saves, measurements should be saved to RAM disk. From RAM disk, they can be copied to the internal floppy disk by pressing (SAVE RECALL) **File Utilities** **Copy All Files** , or using the SCPI command

```
MMEM:COPY 'MEM:*.*', 'INT:'
```

or

```
MMEM:COPY 'RAM:*.*', 'INT:'
```

The following mass storage specifiers can be used:

Table 8-1 **Mass Storage Specifiers**

Disk	SCPI name
Non-Volatile RAM Disk	`MEM:`*file*
Volatile RAM Disk	`RAM:`*file*
Internal 3.5 in Disk	`INT:`*file*

The files can also be transferred over GPIB using the SCPI command `MMEM:TRANsfer`. Refer to the *Example Programs Guide* for details.

Using Hardcopy Features to Print or Plot Results

The analyzer's HARDCOPY feature dumps the measurement display to a printer or plotter in any one of the following formats:

- measurement graph and marker table (default)
- measurement graph only
- marker table only
- trace values at each point only

The HARDCOPY feature can also save the measurement display to a floppy disk in HP-GL, PCX, or PCL5 format. These files can be imported into various computer applications, such as Microsoft Word® or Lotus WordPro®, and integrated with other text and graphics. You can use the SCPI `MMEM:TRANsfer` command to copy files from the analyzer's floppy disk to an external computer. This is described in the *Example Programs Guide*.

HP-GL and PCL5 format files can also be archived on the analyzer's floppy disk drive, and later sent to a printer or plotter. Under program control, the files can be printed using various page layouts, such as one to a page, two to a page, and so on, using either portrait or landscape orientation. This is done using the HP-GL `IP` command, described later.

The analyzer provides an GPIB interface, a Centronics parallel interface, and a LAN interface, all of which are well suited for printing. If you are controlling the analyzer via the GPIB port, you can use the parallel port or LAN port for hardcopy. Or you can have your computer collect the measurement results, format them, and dump them to its own printer.

If you have a custom interface circuit connected to the analyzer's parallel port, you can still connect a printer in addition, and use the printer-select line to select either the printer or your custom interface circuit. For more details, refer to "Writing to the Parallel Port" on page 6-6.

Faster Hardcopies using a PCL5 Printer

If your printer supports PCL5, be sure to select PCL5 format under (HARDCOPY) **Select Copy Port** . PCL5 uses HP-GL format, resulting in very fast, high-resolution hardcopies. PCL5 can result in a speed improvement of up to 10 times, compared to using raster formats such as PCL, Epson-compatible, or PCX. Since the analyzer measurement speed decreases while performing hardcopy rasterization, using PCL5 format will restore your measurements to full speed more quickly.

Some printers that support PCL-5 include:

- HP LaserJet 4, 5
- HP Color LaserJet

Customizing Page Layout

Custom hardcopy output, such as 2 or 4 plots per page, can be done using a PCL5 printer and IBASIC or SCPI commands. To do this you should:

1. Put the printer into HP-GL mode.
2. Send the printer an "IP" HP-GL command.
3. Perform a hardcopy dump in HP-GL format.

These steps are discussed in detail below.

Put the Printer into HP-GL Mode

In order to send HP-GL hardcopy output to your PCL-5 printer, you must first instruct the printer to accept HP-GL commands. On some printers, this can be done using the printer's built-in menu. You can also send the printer a PCL-5 escape sequence to instruct it to accept HP-GL. This can be done using IBASIC or SCPI commands. Refer to your printer manual for details. Refer, also, to the example program titled "FAST_PRT" on the *Example Programs Disk*. This program configures your PCL5 printer to accept HP-GL commands.

Send the Printer an GPIB Command

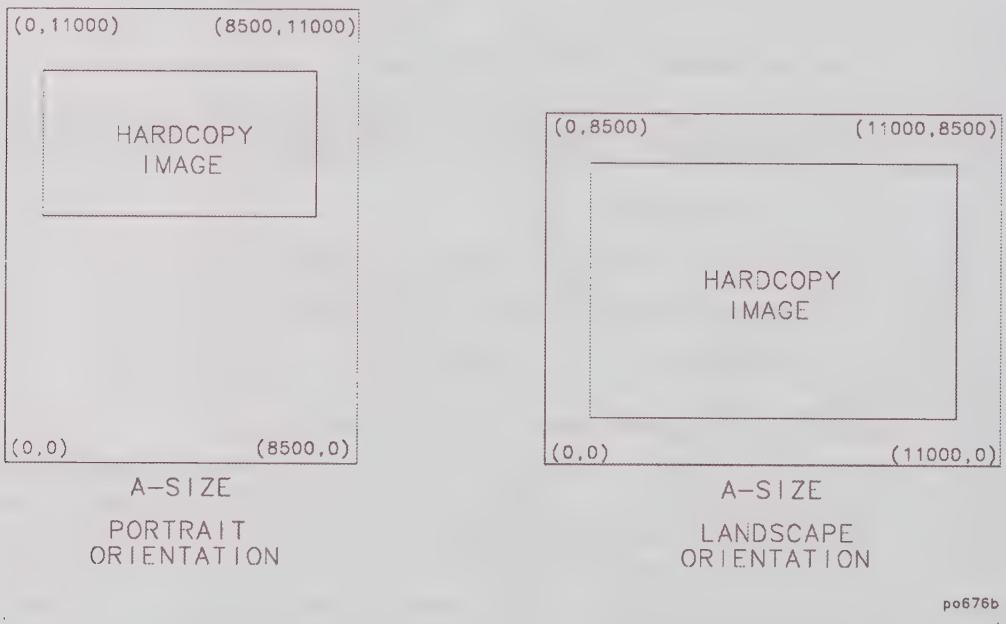
You will need to send the printer an `IP` command to specify a specific rectangular region on the paper. This is done by sending the printer the `IP` HP-GL command.

The `IP` command specifies the size and the position of the printed image on the paper. The units are thousandths of an inch, so 8500 units would be 8.5 inches. The arguments to the IP command are:

`IP` *LowerLeftX,LowerLeftY,UpperRightX,UpperRightY;*

The paper is numbered as shown in Figure 8-1.

Figure 8-1 **Paper Numbering**



Typically, a margin around the image of 0.5 to 1.0 inches (500 to 1000 units) is used.

TIP

When performing hardcopies of the measurement graph only (excluding the marker table), the hardcopy image looks best with an aspect ratio of approximately 1.3:1 (x:y).

Example IP command

For example, to print a hardcopy image on the top of a portrait A-size sheet of paper, you can use about 7.5 inches of the paper's 8.5 inch width. The width of the image would be 7500 units, beginning at the left margin of 500 units and ending at the right margin of 8000 units. Using an aspect ratio of 1.3:1, the height should be 7500 / 1.3 = 5769. The top margin of the paper is at 10000, so the bottom of the image should be at 10000 – 5769 = 4231. Plugging these numbers into the HP-GL `IP` command gives:

```
IP 500,4231,8000,10000;
```

Using the same calculations for an A-size sheet of paper in landscape orientation gives:

```
IP 500,7,10500,7700;
```

The numbers shown in these examples were developed using an HP LaserJet 4. Other printer's margins may vary slightly.

Perform a hardcopy dump HP-GL format

Once your printer is set to accept HP-GL commands, you can perform a hardcopy using the SCPI `HCOP;*WAI` command.

Custom Data Sheets

You can write programs to set the printer font, output text to the printer, and send hardcopy of measurement test results to the printer to create a custom data sheet. Refer to the example program called "REPORT" which is included on the *Example Programs Disk*. A data sheet created by the "REPORT" program is shown on the next page.

The example program uses hardcopy output to generate a report with custom text. Five different text fonts are used. The fonts are available for HP LaserJet printers. Refer to your printer manual to modify the example fonts for your printer.

Figure 8-2 Example of Customized Data Sheet

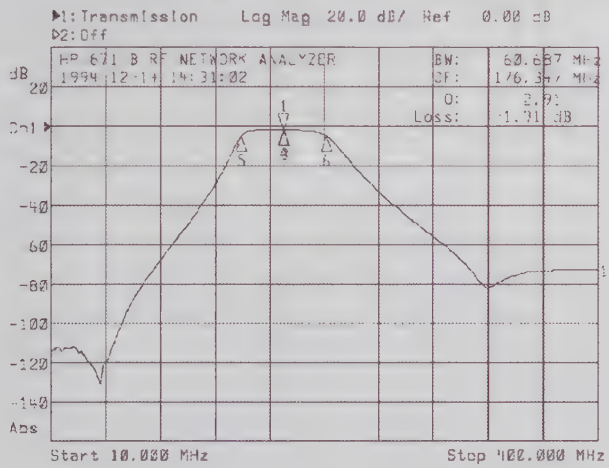
COMPANY NAME
CITY, STATE, COUNTRY

BPF-177 Bandpass Filter

PASS BAND	Level	Bandwidth
	-3 dB	60 MHz +/- 3 MHz
	-20 dB	95 MHz +/- 5 MHz
	-60 dB	200 MHz +/- 8 MHz

SWR PASSBAND (typical)	1.5:1
Cost per unit:	$24.95

Transmission Characteristics



IN STOCK! IMMEDIATE DELIVERY!

For more information: Call 1-800-Filter

po667b_c

Statistical Process Control

If you collect data on your production process, you can use statistics to control and improve your process. Tools such as histograms, Pareto diagrams, and scatter plots can help you to quantify your process' behavior, and identify trends, cycles, and other "unnatural" patterns.

You can purchase computer programs such as SAS and SPlus to perform statistical analyses. HP VEE, which you can use to control your analyzer, also offers some statistical capability. You can also use add-in macros for popular spreadsheet programs.

For example, you can use Microsoft Excel from the Office97 suite to import *.prn (Lotus 1-2-3 format files) directly from the analyzer into an Excel spreadsheet. You can then use the charting features to plot statistical results.

For computers connected to a LAN, the *.prn files can be imported over the LAN directly from the analyzer's internal dynamic data disk. See *The LAN Interface User's Guide Supplement* for information on accessing the analyzer's dynamic data disk.

Transferring Files

Files can be transferred to and from the analyzer via the GPIB or LAN ports. Using the LAN port may be the most convenient if you are transferring files using a UNIX workstation or a PC equipped with Windows95 (or later) and ftp networking software.

Two example programs ("GETFILE" and "PUTFILE") demonstrate how to transfer files from the analyzer's mass memory to and from mass memory of an external controller via GPIB. Instrument states and program files may be transferred to or from the analyzer's internal non-volatile memory, `MEM`, internal-volatile memory, `RAM`, and the internal 3.5 inch floppy disk, `INT`. This can be a convenient method to archive data and programs to a central, large mass-storage device.

These example programs are found on the *Example Programs Disk* that was shipped with your analyzer, and are described and listed in the *Example Programs Guide*. To run these programs, connect an external controller to the analyzer with an GPIB cable.

For information about transferring files over a LAN, see *The LAN Interface User's Guide Supplement*.

Index

Symbols

*LRN, 5-6

A

automation, 1-2, 8-15
  selecting a controller, 2-9
AUTOST files, 2-3, 5-15
Autost key, 5-15
AUTOST program, 3-10
auto-step, 3-11, 4-5

B

barcode reader, 3-13
button box, 3-19

C

C
  programming language, 2-12
computer
  selecting for automation, 2-9
computer connections, 2-7
configurations
  of system for automation, 2-2
connecting analyzers to a computer, 2-7
controller
  selecting for automation, 2-9
controller connections, 2-7
customizing page layout, 8-9

D

diagrams
  on-screen, 3-5
disk access time, 5-5
display
  split, 3-5
DRAW
  graphics keyword, 3-5
driver, VXI plug&play, 2-13

E

example programs, locations, 1-2
external keyboard, 3-14
external keyboard hot keys, 3-16
external monitor, 3-24

F

fast recall, 4-1
foot switch, 3-19

G

graphics, 3-5

H

hardcopy
  faster, 8-9
hot keys
  external keyboard, 3-16
HP BASIC
  programming language, 2-11
HP VEE
  programming language, 2-12

I

IBASIC
  programming language, 2-11
IBASIC and automation, 2-3
instrument states
  recalling, 5-4
iterative control, 5-8

K

keyboard
  external, 3-14
keyboard template, 3-16
keyboard, external
  hot keys, 3-16
keystroke recording, 3-10

L

language
  programming, 2-11
layout, page, 8-9
learn strings, 5-6
limit test port
  used as general purpose I/O, 3-21
limit testing, 3-21
LRN. *See* *LRN

M

macro recording, 3-10
memory, 5-4
message string, 3-4
modifying a program, 3-10
monitor
  external, 3-24
MOVE
  graphics keyword, 3-5

N

networking, 2-7
non-volatile RAM disk, 5-4

O

on-screen diagrams, 3-5

P

page layout, customizing, 8-9
parallel port, 6-2, 6-9
plug&play driver, 2-13
port
  parallel, 6-2, 6-9
printing and plotting, 8-9
programming languages, 2-11
prompting
  how to display pop-up messages, 3-4

Index

R

RAM disks, 5-4
readable ports, 3-23, 6-4
recall from a disk or memory, 4-1
recall, fast, 4-1
recalling states, 4-1, 5-4
recording
  keystroke, 3-10

S

serial port, 6-11
SICL, 2-12
single sweep, 5-12
softkey auto-step, 3-11, 4-5
S-parameters, quick, 5-12
split display, 3-5
SRQ, 5-10
states
  recalling, 5-4
string
  message, 3-4
strings
  learn, 5-6
sweep, single, 5-12
switch
  foot, 3-19
system
  automated, configuration, 2-2

T

template
  keyboard, 3-16
throughput
  of an automated system, 2-8
time
  disk access, 5-5
title feature, 3-15
topology
  measurement system, 2-2
TTL signals, 3-19

U

User BEGIN, 3-6
  default program, 3-8
User BEGIN program
  to load, 3-10
User BEGIN program structure, 3-7
USER TTL IN/OUT port, 3-18

V

VEE
  programming language, 2-12
volatile RAM disk, 5-4
VXI plug&play driver, 2-13

W

writeable ports, 3-22, 6-5